



Aalto-yliopisto
Insinööritieteiden
korkeakoulu

**Sovelluskehitystyökalun valinta mobiilin paikkatietosovelluksen toteuttamiseen
yritysympäristössä**

Aalto-yliopiston insinööritieteiden korkeakoulun
maankäyttötieteiden laitoksella tehty diplomityö

Espoo, tammikuu 2013

Tekniikan kandidaatti Kimmo Jokinen

Valvoja: Professori Kirsi Virrantaus
Ohjaaja: Tekniikan tohtori Jussi Nikander

Aalto-yliopisto
Insinööritieteiden korkeakoulu
Maankäyttötieteiden kirjasto

Sisältö

1	Johdanto	1
2	Tutkimuskohde ja tutkimuskysymykset	3
3	Teoria	5
3.1	Mobiilisovellus ja sen erityispiirteet	5
3.1.1	Mobiilin tietojenkäsittelyn haasteet	6
3.1.2	Sovelluskehitys mobiililaitteille	8
3.1.3	Mobiilikäyttökonteksti	10
3.2	Mobiilisovellustyypit	11
3.2.1	Mobiili-web-sivu	12
3.2.2	Natiivisovellus	12
3.2.3	Web-sovellus	12
3.2.4	Hybridisovellus	13
3.3	Mobiili paikkatietosovellus	13
3.4	Toiminnanohjausjärjestelmä	15
3.4.1	SAP	16
3.5	Mobiili yritysjärjestelmä	17
3.5.1	Mobiili SAP	19
3.6	Paikkatieto yritysjärjestelmissä	21
4	Mobiilin SAP-paikkatietosovelluksen toiminnallisuudet	22
4.1	Mobiilin paikkatiedon sovellustoiminnallisuudet	22
4.1.1	Kartta	22
4.1.2	Paikannus	23
4.1.3	Navigointi	23
4.1.4	Paikkatietoanalyysit	24
4.2	SAP:n mobiilisovellustoiminnallisuudet	25
4.2.1	Yhdistettävyys SAP-dataan	25
4.2.2	Käyttöoikeuksien hallinta	26
4.2.3	Tiedonhallinta mobiililaitteella	27
5	ISO-standardit ohjelmiston laatuun liittyen	28
5.1	ISO/IEC 9126 Toiminnallisuus	31
5.2	ISO/IEC 9126 Luotettavuus	31
5.3	ISO/IEC 9126 Käytettävyys	31
5.4	ISO/IEC 9126 Tehokkuus	32
5.5	ISO/IEC 9126 Ylläpidettävyys	32
5.6	ISO/IEC 9126 Siirrettävyys	33

6	Analyysimenetelmä	34
6.1	Analyysikriteeristöä yleisesti.....	34
6.2	Toiminnallisuuden kriteerit.....	35
6.3	Luotettavuuden kriteerit.....	36
6.4	Käytettävyyden kriteerit.....	37
6.5	Tehokkuuden kriteerit	39
6.6	Ylläpidettävyyden kriteerit.....	40
6.7	Siirrettävyyden kriteerit	41
7	Aineisto.....	43
7.1	SAP UI Development Toolkit for HTML5	43
7.2	Adobe PhoneGap	44
7.3	Appcelerator: Titanium Development Platform.....	45
7.4	Sencha Touch	45
7.5	Sybase Mobile SDK.....	46
8	Aineiston käsittely ja tulokset.....	47
8.1	Sovelluskehysten toiminnallisuus-laatu	47
8.1.1	Soveltuvuus.....	47
8.1.2	Tarkkuus	50
8.1.3	Yhteensopivuus.....	51
8.1.4	Turvallisuus	51
8.2	Sovelluskehysten luotettavuus-laatu	54
8.3	Sovelluskehysten käytettävyys-laatu	56
8.4	Sovelluskehysten ylläpidettävyys-laatu	60
8.5	Sovelluskehysten siirrettävyys-laatu	63
8.5.1	Sopeutuvuus.....	63
8.5.2	Asennettavuus	64
8.5.3	Korvattavuus.....	66
9	Tulosten tarkastelu	69
9.1	Tulokset laatuominaisuuksien näkökulmasta.....	70
9.2	Johtopäätökset.....	72
10	Yhteenveto	74
	Lähdeluettelo	76



Tekijä Kimmo Jokinen

Työn nimi Sovelluskehitystyökalun valinta mobiilin paikkatietosovelluksen toteuttamiseen yritysympäristössä

Laitos Maankäyttötieteiden laitos

Professuuri Kartografia ja geoinformatiikka

Professuurikoodi Maa-123

Työn valvoja Professori Kirsi Virrantaus

Työn ohjaaja(t) Tekniikan tohtori Jussi Nikander, DI Markku Matilainen

Päivämäärä 31.1.2013

Sivumäärä 81

Kieli Suomi

Tiivistelmä

Tässä diplomityössä selvitetään sovelluskehitystyökalun valintaan liittyviä tekijöitä mobiilin yritys ympäristön paikkatietosovellusten kontekstissa. Tavoitteena on kuvata valintapäätöksen kannalta merkittävimmät laatuominaisuudet ja -kriteerit, joiden perusteella sovelluskehityksiä voidaan verrata keskenään. Tutkimuksessa toteutetaan myös vertailu SAP:n tukemien mobiilisovelluskehysten välillä, minkä pohjalta voidaan arvioida, mikä sovelluskehityksistä on parhaiten sopiva SAP-ympäristössä käytettäväksi. Tutkittavat sovelluskehitykset ovat: SAP UI Development Toolkit for HTML5 (SAP UI5), Adobe PhoneGap, Appcelerator: Titanium Development Platform (Appcelerator), Sencha Touch sekä Sybase Mobile SDK.

Tutkimuksessa luodaan ISO/IEC 9126 laatumalliin pohjautuva kriteeristö, jonka avulla sovelluskehysten laatua voidaan arvioida ja niitä voidaan vertailla keskenään. ISO/IEC 9126 jakautuu kuuteen laatuominaisuuteen ja näiden alaominaisuuksiin. Olennaisimmiksi arvioiduille ja staattisesti mitattavissa oleville alaominaisuuksille määritellään arviointikriteerit.

Tutkimuksen perusteella mikään sovelluskehityksistä ei ole selkeästi muita parempi, vaan useat tekijät, kuten tuettujen mobiilialustojen määrä sekä tarjotut sovellustoiminnallisuudet, vaikuttavat paremmuuteen. Esimerkiksi Sybase Mobile SDK tarjoaa eniten yrityssovelluksissa tarvittavia yleisiä toiminnallisuuksia, kun taas Adobe PhoneGap mahdollistaa sovelluskehityksen laajimmalle määrälle eri mobiilialustoja. Sybase Mobile SDK:lla haittapuolena on esimerkiksi, että monialustakehityksessä ei voida hyödyntää samaa sovelluskoodia eri alustoilla. Tässä suhteessa taas esimerkiksi Appcelerator ja Sencha Touch ovat parempia.

Sovelluskehityksistä erottuu kuitenkin yksi sovelluskehitys, SAP UI5, jota ei voida suositella käytettäväksi. Sitä ei voida suositella, koska se ei tarjoa vielä ratkaisua mobiilikehitykseen. Tuki mobiilikehitykseen on sille kuitenkin luvattu.

Johtopäätöksenä parhaimman sovelluskehityksen arviointi ei ole yksinkertaista. Valintapäätöksen tueksi voidaan kuitenkin vertailla useita vaihtoehtoja määriteltujen kriteerien avulla. Kriteerien avulla voidaan arvioida sovelluskehysten hyviä ja huonoja puolia. Arviot voidaan suhteuttaa tämänhetkisiin ja tulevaisuuden sovellustarpeisiin, minkä perusteella sovelluskehitys tai -kehitykset voidaan valita.

Avainsanat Mobiilius, sovelluskehitys, paikkatieto, yrityssovellus, laatu, ISO/IEC 9126

Author Kimmo Jokinen

Title of thesis Framework selection for developing mobile geographic information applications in enterprise context

Department Department of Real Estate, Planning and Geoinformatics

Professorship Cartography and Geoinformatics

Code of professorship Maa-123

Thesis supervisor Professor Kirsi Virrantaus

Thesis advisor(s) Doctor of Science (Technology) Jussi Nikander, M.Sc. Markku Matilainen

Date 31.1.2013

Number of pages 81

Language Finnish

Abstract

This thesis examines the selection of a mobile software framework for the development of enterprise geographic information applications. The aim of the study is to describe the most important quality characteristics with which frameworks can be compared with each other. The purpose of the comparison is to find the most suitable framework for use in SAP environment. Analysed frameworks are: SAP UI Development Toolkit for HTML5 (SAP UI5), Adobe PhoneGap, Appcelerator: Titanium Development Platform (Appcelerator), Sencha Touch, and Sybase Mobile SDK.

Evaluation criteria for the comparison are based on the ISO/IEC 9126 software quality model. ISO/IEC 9126 is divided into 6 quality characteristics and their sub-characteristics. Evaluation criteria are presented for the most important and statically measurable sub-characteristics.

Based on this study, no framework is clearly better than the others. Many factors, such as the mobile platforms supported and the features provided, affect the quality of a framework. For example, common enterprise application features are best provided by Sybase Mobile SDK, whereas Adobe PhoneGap allows development for the highest number of different mobile platforms. One downside of Sybase Mobile SDK is that the same source code cannot be used on different platforms. In that respect, Appcelerator and Sencha Touch, among others, are better.

However, one framework, SAP UI5, is not suited for mobile development. It cannot be recommended because it does not provide support for mobile development. However, support for mobile development has been promised for it.

In summary, finding the most suitable software framework is not easy. However, the evaluation criteria defined in this work can be used to compare different alternatives. The criteria enable assessing the important advantages and disadvantages of the frameworks in a structured way. Evaluations can then be compared to current and future application needs. This way the most applicable framework or frameworks can be selected.

Keywords Mobile context, software framework, geographic information, enterprise application, quality, ISO/IEC 9126

Esipuhe

Tämä diplomityö on tehty Neomore Consulting:lle pitkälti syksyn 2012 aikana. Olen työskennellyt yrityksessä vuodesta 2009 alkaen ja jo pitkään ennen työn aloittamista olimme sopineet, että teen diplomityöni yritykselle, mikäli löydämme sopivan aiheen.

Viimein alkuvuodesta 2012 ajatus diplomityön tekemisestä konkretisoitui ja aloimme yhdessä Neomore:n Erkkä Tahvanaisen, Kristian Rantakosken, Markku Matilaisen ja Tuomas Haimin kanssa miettiä tarkempaa aihetta. Sopivan aiheen löytämisessä menikin aikansa. Tämä pitkälti sen takia, että yritys tarjoaa SAP:iin liittyvää konsultointia ja SAP:n yhdistäminen omaan pääaineeseen geoinformatiikkaan ei heti tarjonnut luontevaa aihepiiriä tutkimukselle. Lopulta molempia osapuolia tyydyttävä aihe löytyi. Tahdonkin sanoa kiitoksen Erkalle ja Kristianille mahdollisuudesta tehdä diplomityöni yritykselle. Kiitokset myös, ettette häirinneet minua liiaksi muilla töillä niinä päivinä, kun tein diplomityötä. Tuomakselle myös kiitokset hyvistä ajankohtaisista huomioista työn aikana.

Erityiskiitoksen ohjauksesta haluan antaa tohtori Jussi Nikanderille. Säännölliset tapaamiset hänen kanssaan auttoivat minua selvittämään isoimmat ongelmakohdat työssä. Pitkälti Jussin ansiosta työ myös valmistui lopulta nopeammin kuin itsekään olin uskaltanut toivoa.

Kiitokset myös Marialle tärkeästä kannustuksesta diplomityön teon aikana sekä aidosta kiinnostuksesta diplomityötäni ja erityisesti sen kieliasua kohtaan.

Espoossa 19. Tammikuuta 2013

Kimmo Jokinen

1 Johdanto

Mobiililaitteiden ja -sovellusten käyttö on viime vuosina kasvanut räjähdysmäisesti ja kasvun on ennustettu jatkuvan voimakkaana myös tulevaisuudessa (Statista 2012; Strategy Analytics 2012). Mobiilikäyttö on tullut työpöytäkäytön rinnalle monissa tilanteissa. Ihmiset ovat vapaa-ajallaan tottuneet käyttämään älypuhelimiaan monien päivittäisten askareiden tekemiseen, joten myös yrityksissä mahdollisuus mobiilien sovellusten käyttämiselle on tullut luontevammaksi. Työn tekeminen onkin nykyisin aikaisempaa vähemmän sidottua fyysiseen työpaikkaan. Työtä voidaan tehdä työpaikan lisäksi esimerkiksi kotona, asiakkaiden luona, työmatkoilla ja mökin laiturilla.

Eri sijainneissa tehtävä työ asettaa omat vaatimuksensa ja tarpeensa työlle ja niiden konteksti vaikuttaakin olennaisesti tehtävään työhön. Työpaikalla on esimerkiksi paremmin aikaa täyttää pitkiä lomakkeita, kuin mitä on asiakkaan luona myyntitehtävissä vieraillessa. Työn teon jakautuessa eri paikkoihin tulee olennaisemmaksi työntöön kontekstin huomioiminen. Tämä voidaan saavuttaa muun muassa toteuttamalla sovellusten käyttöliittymät kontekstin perusteella muokkautuviksi. Esimerkiksi myyntilaskulle voidaan täydentää automaattisesti joitain tietoja, kuten asiakkaan nimi ja osoite, sijainnin perusteella. Kontekstin huomioiminen onkin olennainen osa sovelluksen käyttötapauksia mietittäessä. Ei ole syytä olettaa, että mobiilisovellusta käytetään täysin samaan tarkoitukseen kuin työpöytäsovellusta, vaan se voidaan esimerkiksi suunnitella täyttämään ainoastaan tärkeimmät liikkuvan työn käyttötapaukset. (Aberdeen Group 2000)

Yrityksen toiminnanohjausjärjestelmällä ja mobiileilla sovelluksilla on tärkeää tehostaa prosesseja, jotta yritys pärjäisi paremmin (Aberdeen Group 2000). Esimerkiksi mitä nopeammin myyntilaskut saadaan täytettyä, sitä enemmän työntekijöillä jää aikaa muulle työlle. Paikkatiedon tarjoama kontekstin huomioiminen mahdollistaa myyntilaskun nopeamman täyttämisen, kun osa laskun kentistä voidaan täydentää automaattisesti. Paikkatiedon voikin nähdä olennaisena osana mobiilia työkenttää ja sen yhdistämisellä muuhun yrityksen tietoon on saavutettavissa merkittäviä hyötyjä.

Yritysjärjestelmillä on toiminnan tehostamisen lisäksi useita tärkeitä ominaisuuksia, kuten integroitavuus, suojattavuus ja jatkokehittävyyys. Järjestelmiä joudutaan integroimaan keskenään esimerkiksi ottaessa uusia järjestelmiä ja sovelluksia käyttöön. Yrityksen järjestelmien suojaus täytyy olla ajan tasalla, jotta ainoastaan autorisoiduilla henkilöillä on oikeus tiettyihin tietoihin. Autorisoinnit eivät kuitenkaan saisi optimitilanteesta estää tai hidastaa työn tekemistä. Esimerkiksi kertakirjautuminen on kehitetty vähentämään muistettavien salasanojen ja manuaalisten kirjautumisten määrää, jotta työn tekeminen tehostuisi. Jatkokehittävyydellä tarkoitetaan sovellusten tehokasta mukauttamista yrityksen alati muuttuviin tarpeisiin. Toiminnanohjausjärjestelmät, kuten SAP¹, pyrkivät mahdollistamaan esimerkiksi näiden ominaisuuksien huomioon ottamisen mahdollisimman tehokkaasti. (Aberdeen Group 2000. Al-Mashari ym. 2002)

Esimerkiksi integroitavuus ja suojattavuus ovat olennaisia tarpeita myös yrityksen mobiilisovellusten osalta. SAP AG (jatkossa SAP) tunnistaa mobiilisovellusten tarpeen nostamalla mobiiliuden ”ajankohtaiseksi aiheeksi”(SAP 2012a). Sillä ei kuitenkaan ole tällä hetkellä tarjota yksittäistä sovelluskehitystyökalua tai sovellusta, jolla nämä tarpeet

¹ SAP on laajasti käytössä oleva toiminnanohjausjärjestelmä. Viitattu 23.11.2012. Saatavissa: <http://www.sap.com>.

saataisiin täytettyä. Sen sijaan SAP tukee useita eri vaihtoehtoisia sovelluskehitystyökaluja, joilla mobiilit liiketoiminta- ja sovellustarpeet voidaan täyttää. Osa sovelluskehitystyökaluista on kuitenkin vielä kehitysasteella ja loputkaan eivät ole vielä vakiinnuttaneet itseään käytössä. Nämä asiat yhdistettynä muutenkin nopeaan mobiililaitteiden ja työkalujen kehityksen tahtiin vaikeuttaa sovelluskehitystyökalun valintaa paikkatietoon ja SAP:iin linkittyvien mobiilisovellustarpeiden toteuttamiseen.

Sovelluskehysten, ja ylipäänsä ohjelmistojen, valinnassa korostuu laadun käsite. Miten todennetaan, että sovelluskehukset ovat laadukkaita ja esimerkiksi tukevat riittävällä tavalla liiketoimintaprosessien toteuttamista. Ohjelmiston laatu itsessään on jo hyvin laaja käsite ja sille ei ole yksikäsitteistä ja yleisesti hyväksyttyä määritelmää, vaan määritelmiä on useita erilaisia. Muun muassa Wong ja Jeffery (2001) kuvaavat laajasti laadun eri määritelmiä sekä niiden yhteisiä ja eriaviä ominaisuuksia. Laadun mittaaminen tai varmentaminen ei myöskään ole yksinkertaista, vaan siihen vaikuttaa laadun määritelmän lisäksi, mitä laatuominaisuutta ohjelmistosta halutaan mitata. Vaihtoehtoisten tuotteiden valinnassa laatumalli onkin syytä määritellä yhtenäisiksi eri tuotteiden välillä sekä valita laadulle mittarit, jotka kuvaavat tarpeita mahdollisimman kattavasti.

Tutkimuksessa vertaillaan, miten eri SAP:n suosittelemat sovelluskehukset soveltuvat mobiiliin yritys ympäristön paikkatietosovelluksen toteuttamiseen. Tarkasteltavat sovelluskehukset ovat: SAP UI Development Toolkit (SAP UI5), Adobe PhoneGap, Appcelerator: Titanium Development Platform, Sencha Touch sekä Sybase Mobile SDK. Tavoitteena on selvittää erottuuko jokin sovelluskehyksistä edukseen mobiiliin yritys ympäristön paikkatietosovellusten kehityksen kannalta.

Työ rakentuu siten, että johdannon jälkeen luvussa 2 esitellään tutkimuskohde sekä tutkimuskysymykset. Tutkimukseen liittyvää teoriaa käsitellään luvuissa 3-5, joista luvussa 3 tarkastellaan mobiilisovellusten teoriaa yleisemmällä tasolla, kun taas luvussa 4 keskitytään tärkeimpiin mobiilisovellustoiminnallisiin, jotka on johdettu luvun 3 pohjalta. Luvussa 5 esitellään analyysissä käytettävä ISO/IEC 9126-laatumalli (2001). Luku 6 kuvaa tutkimuksessa käytettävän analyysimenetelmän sekä -kriteeristön. Luvussa 7 esitellään tutkittavat sovelluskehukset. Luvussa 8 kuvataan tutkimuksen analyysiosuus ja analyysin tulokset. Luvussa 9 tiivistetään analyysin tulokset ja esitetään vastaukset tutkimuskysymyksiin. Lopuksi luvussa 10 arvioidaan tutkimusta esimerkiksi sen hyödyllisyyden näkökulmasta sekä ehdotetaan ideoita jatkotutkimukselle.

2 Tutkimuskohde ja tutkimuskysymykset

Tutkimuksessa vertaillaan, miten eri SAP:n suosittamat käytössä olevat sovelluskehitysteknologiat soveltuvat mobiilin paikkatieto-SAP -sovelluksen toteuttamiseen. Sovelluskehitysteknologioita vertaillaan teoriasta (mobiili, konteksti, viitekehys) johdettujen yleiseksi arvioitujen paikkatietoon ja SAP:iin liittyvien käyttötapausten osalta. Vertailun pohjalta pyritään tukemaan sovelluskehitysteknologian valintaa mobiilin paikkatieto-SAP -sovellustarpeen toteuttamiseen, koska sopivan teknologian valinnan on arvioitu olevan tällä hetkellä vaikeaa sovelluskehitysteknologioiden suuresta määrästä sekä keskinäisistä eroista johtuen.

Tutkimuskysymykset:

1. Mitkä ovat tärkeimmät sovellusominaisuudet mobiilia SAP-paikkatietosovellusta toteutettaessa?

Tärkeimmät sovellusominaisuudet joko toistuvat sovelluksesta riippumatta tai ovat ainakin useimmissa sovelluksissa läsnä. Näin ollen sovelluskehitystyökalun mahdollistamien sovellusominaisuuksien arvioidaan olevan merkittävä päätökseen vaikuttava tekijä työkalua valittaessa.

2. Mitkä ominaisuudet määrittelevät, mikä sovelluskehitys soveltuu laadullisesti parhaiten mobiiliin SAP-paikkatietoympäristön sovellukseksi?
 - o Miten sovelluskehysten laatu ilmenee eri laatuominaisuuksien osalta? Onko esimerkiksi jokin sovelluskehitys käytettävyyden kannalta laadukkaampi kuin jokin toinen?

Sovelluskehysten laatua voidaan arvioida monella menetelmällä tai kriteerillä. Arviossa voidaan myös painottaa eri näkökulmia, kuten työkalun laajuutta tai päivitystiheyttä. Mobiilikehitykseen sovelluskehystä valittaessa onkin tärkeää, että niitä arvioidaan ja vertaillaan perusteltujen ja kontekstiin soveltuvien menetelmien ja kriteerien avulla painottaen mobiiliuden kannalta olennaisimpia laatuominaisuuksia.

3. Mikä SAP:n suosittamista sovelluskehyksistä on laadukkain mobiiliin SAP-paikkatieto -sovelluksen toteuttamiseen?
 - o Esimerkiksi onko jokin sovelluskehitys laadukkaampi paikkatiedon kannalta tai onko jokin toinen taas vastaavasti laadukkaampi SAP-integraation kannalta?

Mobiiliin SAP-ympäristöön sovelluskehitystyökalua valitessa on tärkeää tutkia, miten sovelluskehukset eroavat laatuominaisuuksien osalta toisistaan, jotta voidaan valita kontekstiin ja sovellustarpeiden täyttämiseen soveltuva työkalu. Erilaisten sovellusominaisuuksien huomioiminen on myös tärkeää, jotta voidaan arvioida sovelluskehysten vahvoja ja heikkoja puolia eri näkökulmista.

4. Onko tutkimuksen pohjalta perusteltua, että SAP ei tue vain jotain tiettyä sovelluskehystä?

Yksittäistä sovelluskehystä tukemalla sovelluskehitystyökalun valinta helpottuisi sovelluskehittäjän kannalta. Toisaalta sitoutuminen tiettyyn työkaluun voisi aiheuttaa ongelmia esimerkiksi yleisten sovellusalojen muuttuessa.

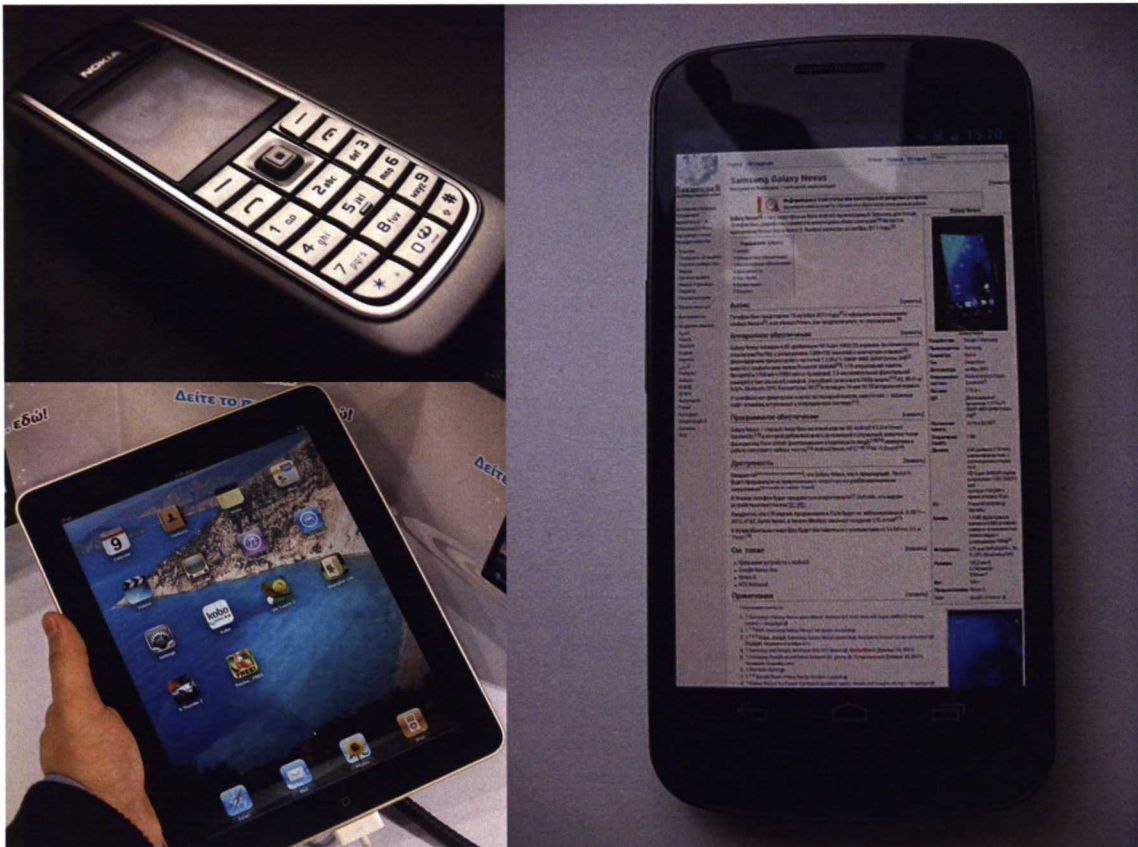
SAP tukee tällä hetkellä useita eri mobiilisovelluskehitysteknologioita, eikä se tarjoa juurikaan apua niiden valintaan sovellustarpeiden näkökulmasta. SAP ei myöskään ole tarkasti täsmentänyt yksittäisten mobiilisovelluskehitystyökalujen osalta, miten se tulevaisuudessa tukee ja mahdollisesti integroi niitä laajemmin SAP-ympäristöön, vaan se lähinnä tarjoaa useita vaihtoehtoja (Moy 2012). Mobiilisovelluskehitystyökalut myös kehittyvät hyvin nopeasti ja uusia työkaluja tulee jatkuvasti lisää. Myöskään esimerkiksi mobiilisovellusten kehittämiseen vaikuttavat HTML5 (W3C 2012a) ja siihen kuuluvat spesifikaatiot (mm. W3C 2011; W3C 2012b; W3C 2012c) eivät ole vielä valmiita. SAP AG:n näkökulmasta on ymmärrettävää, että tässä epävarmassa tilanteessa he eivät halua sitouttaa itseään ainoastaan yhteen tekniikkaan, mutta sovelluskehitystyökalun valitsijan näkökulmasta vaihtoehtojen paljous asettaa haasteita. Tutkimus pyrkii vastaamaan onko perusteltua, että SAP ei halua tehdä valintaa yksittäisen sovelluskehityksen puolesta, vaan pyrkii tarjoamaan näennäisesti toisiaan vastaavia vaihtoehtoja? Yritysympäristössä sovellusten ja varsinkin niiden tukemien liiketoimintaprosessien elinkaari on usein pitkä (Lehman 1980; Kraftig ym. 2004), joten sovelluskehitystyökalua valittaessa on syytä arvioida, miten työkalu pystyy palvelemaan sovelluksen kehitystä koko sen elinkaaren ajan.

Mobiilikäyttö eroaa työpöytäkäytöstä usealla eri tavalla. Mobiilisovelluksia käytetään usein lyhyempiä aikoja kerrallaan, mutta käyttökertoja voi olla päivässä useampia. Työpöytäsovelluksia taas käytetään usein pidempiä aikoja kerrallaan. Mobiilisovelluksilla käyttöympäristössä voi myös olla enemmän häiriötekijöitä kuin työpöytäsovellusten käyttöympäristössä (Oinas-Kukkonen & Kurkela, 2003). Tutkimuksessa pyritäänkin selvittämään, mitkä ovat mobiilissa yritysympäristössä olennaisimpia sovellusominaisuuksia tyypillisten käyttötapausten ja teorian kautta. Tutkimus pyrkii myös vastaamaan, miten eri SAP-ympäristön mobiilisovelluskehityksillä voidaan, esimerkiksi sovellusominaisuuksien toteutettavuuden kannalta, tarjota laadukkaita sovelluksia käyttäjille. Tutkimuksessa selvitetään myös, onko sovelluskehysten välillä suuria eroja laadun eri osa-alueiden kannalta. Tutkimuksessa arvioidaan, onko jokin sovelluskehys esimerkiksi laadultaan parempi käytettävyyden kannalta kuin jokin toinen sovelluskehys, joka taas voi olla tarjolla olevien toiminnallisuuksien kannalta laadultaan parempi.

3 Teoria

3.1 Mobiilisovellus ja sen erityispiirteet

Mobiilisovellus on sovellus, joka ajetaan mobiililaitteella, kuten ominaisuuspuhelimella (feature phone), älypuhelimella (smart phone), tabletilla tai kämmentietokoneella. Mobiililaitteelle ominaista on, että se on mukana kannettavan kokoinen sekä toimii akulla. Muut näistä ovat nykyisin laajalti käytössä, lukuun ottamatta kämmentietokonetta, jonka tarpeen vastaavat ominaisuudet tarjoavat älypuhelin ja tabletti ovat melko lailla korvanneet.



Kuva 1. Erilaisia mobiililaitteita. Ylhäällä vasemmalla on ominaisuuspuhelin (Nokia 6020), oikealla on älypuhelin (Samsung Galaxy Nexus) sekä vasemmalla alhaalla on tabletti (Apple iPad). (Kuvien lähteet samassa järjestyksessä Senna 2005; Sancho85 2012; Karakatsanis 2010).

Ominaisuuspuhelimien ja älypuhelimien välinen rajanveto ei ole tarkka, mutta yleensä älypuhelimiksi luokitellaan puhelimet, joiden suunnitellut käyttökohteet ovat avoimemmat kuin ominaisuuspuhelimilla ja joihin on esimerkiksi mahdollista asentaa uusia sovelluksia. Ominaisuuspuhelimet on taas suunniteltu jotain tiettyä tarkoitusta, kuten kuvaamista, silmällä pitäen ja esimerkiksi asennettavien sovellusten määrä on pieni. Ominaisuuspuhelimissa mobiilisovellusten toteuttaminen on enemmän laitteen valmistajan kuin ulkopuolisten sovelluskehittäjien vastuulla ja ulkopuolisille sovelluskehittäjille saatavilla olevat työkalut eivät välttämättä ole kovin kehittyneet, joten varsinkin laajempien sovellusten tai sovelluskehysten toteuttaminen on tarkoituksenmukaista vain älypuhelimille. Älypuhelimet voidaan nähdä ensimmäisenä varsinaisena sulautetun ja kaikkialla käytettävissä olevan tietojenkäsittelyn laitteena. (Ballagas ym. 2006)

Työpöytätietokoneella taustajärjestelmiin tehtävät kyselyt ovat usein perinteisiä, eli niillä halutaan päästä käsiksi tarvittavaan tietoon antamalla jokin syöte. Perinteisen kyselyn tulokset riippuvat ainoastaan annetusta syötteestä. Mobiililaitteella taustajärjestelmiin tehtävät kyselyt voidaan jakaa perinteisten kyselyjen lisäksi sijainti- ja kontekstiriippuvaisiin (Taniar & Rahayu 2002; Waluyo ym. 2005). Sijaintikyselyt eroavat perinteisistä kyselyistä siten, että syöteenä on myös laitteen sijainti, joka voi olla käyttäjän syöttämä tai laitteen itsestään täydentämä. Sijaintikyselyillä on mahdollista esimerkiksi kysellä palvelimelta lähistöllä sijaitsevia ravintoloita. Kontekstiriippuvalaiset kyselyt taas hyödyntävät esimerkiksi tietoa käyttäjän sen hetkiseen sijaintiin liittyvistä tarpeista. Näin järjestelmä osaa antaa käyttäjän sen hetkistä tarvetta paremmin vastaavan tuloksen. Järjestelmä voi esimerkiksi pidentää tai lyhentää kyselyn vastauksen pituutta sen perusteella, onko käyttäjä omalla rauhallisella istuimella työpaikallaan vai seisooko hän ruuhkaisessa bussissa. (Waluyo ym. 2009)

3.1.1 Mobiilin tietojenkäsittelyn haasteet

Mobiilisovelluksella tarkoitetaan sovellusta, joka ajetaan ja jota käytetään mobiililaitteella eli mukana kannettavalla laitteella, joka on langattomasti yhteydessä tietoverkkoihin (Forman & Zahorjan 1994). Mobiililaitteita ovat esimerkiksi kämmentietokoneet, älypuhelimet ja tabletit. Forman ja Zahorjan määrittelevät kolme olennaisen tärkeää ja yleistä mobiilille tietojenkäsittelylle ominaista ja haasteita asettavaa ominaisuutta: kommunikaatio (communication), liikkuvuus (mobility) ja siirrettävyys (portability). Tehon kasvun myötä nykyisillä mobiililaitteilla on mahdollista toteuttaa pitkälti samoja toiminnallisuuksia kuin työpöytäkoneillakin (Carroll & Heiser 2010). Mobiililaitteilla merkittävä rajoite on kuitenkin akkukesto sekä akun lämpeneminen. (van Berkel 2009)

Kommunikaatio tai langaton kommunikaatio tarkoittaa mobiilin laitteen kykyä olla yhteydessä muihin laitteisiin tai tietoverkkoihin. Kommunikaatioon liittyviä haasteita mobiilissa tietojenkäsittelyssä ovat tietoliikenneyhteyksien katkeaminen, hitaat yhteydet, suuret erot yhteysnopeuksissa ja -verkoissa sekä tietoturvaongelmat (Forman ja Zahorjan 1994). Vaikka vuoden 1994 jälkeen on tapahtunut paljon kehitystä mobiileissa tiedonsiirtotekniikoissa, ei silti kaikkia Formanin ja Zahorjanin esittämiä haasteita olla saatu ratkaistua (Oinas-Kukkonen & Kurkela, 2003). Yhä esimerkiksi mobiiliyhteydet ovat virheherkkiä ja yhteysnopeudet vaihtelevat varsinkin matkapuhelinverkoissa suuresti. Paikoin yhteysnopeudet jäävät reilusti alle 100 kbit/s vasteaikojen ollessa satoja millisekunteja esimerkiksi GPRS-yhteyttä käyttäessä. Tällöin esimerkiksi reaaliaikaisen videokuvan vastaanottaminen laitteella ei välttämättä onnistu.

Mobiiliverkoissa, kuten langallisissakin verkoissa, yhteysnopeudet ovat usein asymmetriset eli yhteysnopeudet laitteesta verkkoon ovat pienemmät kuin verkosta laitteeseen. Tietoliikenneyhteyksien asymmetrisyydestä johtuen onkin esitetty (Barbará 1999), että tiedonsiirron kannalta on parempi kehittää mobiilisovelluksia, joissa palvelin työntää (push) dataa sovelluksille sen sijaan, että kehitettäisiin sovelluksia, jotka kyselevät (pull) dataa jatkuvasti palvelimelta. Tiedon työntö palvelimelta mobiilisovellukselle on parempi kuin tiedon veto myös sen kannalta, että taustajärjestelmä on monissa tapauksissa paremmin perillä, milloin tieto on muuttunut tai päivittynyt niin paljon, että se on syytä lähettää uudessa muodossaan mobiilisovellukselle. Dataa työntäessä haasteena on kuitenkin päättää, mitä dataa työnnetään palvelimelta millekin asiakkaalle varsinkin, mikäli asiakkaiden tai datan määrä on hyvin suuri. Yhteyksien katkeaminen aiheuttaa myös

ongelmia datan työntöä käyttäessä, koska tällöin on riski, että osa tiedosta jää siirtymättä asiakasovellukseen. (Barbará 1999)

Liikkuvuudella tarkoitetaan kykyä vaihtaa sijaintia verkkoon kytkettynä. Esimerkiksi junalla tai autolla matkustettaessa matkapuhelin vaihtaa tietoliikenneyhteyden tukiasemasta toiseen tietyin väliajoin. Matkustajan sijainnin lisäksi myös tietoliikenneosoitteet sekä tietoliikenneyhteyden laatu voivat muuttua tukiasemalta toiselle vaihdettaessa. Tämä asettaa haasteensa varsinkin, mikäli yhteyden olisi syytä olla tasalaatuinen. Liikkuvuuden seurauksena myös esimerkiksi monia asetuksia ei voida olettaa muuttumattomiksi, vaan niiden on päivityttävä tarvittaessa. Tällaisia ovat esimerkiksi saatavilla olevat tulostimet tai oikea aikavyöhyke. Liikkuvuus mahdollistaa kuitenkin myös sijainnin hyödyntämisen. Työpaikan langattomaan verkkoon siirryttäessä matkapuhelin voi esimerkiksi automaattisesti laittaa sähköpostien reaaliaikaisen hakemisen päälle. Liikkuvuus ja sijainnin seuraaminen asettavat omat vaatimuksensa myös yksityisyyden osalta. Sijainnin seuraaminen mahdollistaa työntekijöiden reaaliaikaisen seurannan, jolloin esimerkiksi yrityksen työntekijälleen tarjoaman matkapuhelimen sijaintia on mahdollista seurata myös työntekijän vapaa-ajalla. (Forman & Zahorjan 1994)

Mobiililaitteet eroavat työpöytäkoneista olennaisesti siirrettävyyden osalta. Mobiililaitteet on suunniteltu kuljetettavaksi mukana, kun taas työpöytäkoneita siirretään lähinnä muuttojen yhteydessä. Mobiililaitteilla onkin olennaista, että ne ovat riittävän kevyitä ja pienikokoisia. Siirrettävyyden kannalta yksi tärkeimmistä mobiilille käytölle rajoitteita asettavasti komponenteista on akku. Mobiililaitteille akkuja suunniteltaessa on tasapainoitava akun kapasiteetin ja fyysisten mittojen, kuten painon, leveyden ja korkeuden, välillä ottaen huomioon myös akun lämpenemisen. Jos akun kapasiteettia kasvatetaan, saadaan sillä enemmän työaikaa. Tämä tapahtuu kuitenkin usein kannettavuuden kustannuksella, koska kapasiteetin kasvattaminen edellyttää usein fyysisten mittojen kasvattamista. Jos taas akun kapasiteetti on pieni, täytyy sitä ladata useammin, mikä saattaa esimerkiksi estää laitteen käyttämistä kokonaisen työpäivän ajan, mikäli latauspisteitä ei ole lähettyvillä. Mobiililaitteen lämpeneminen on myös merkittävä rajoite kädessä kannettavien ja tehokkaaseen laskentaan pystyvien älypuhelimien osalta. Yksinkertaisesti kaikkia laitteen komponentteja ei voida käyttää samanaikaisesti täydellä teholla pitkään tai laitteen komponentit, kuten akku ja prosessori, lämpenevät liikaa. (Forman & Zahorjan 1994; van Berkel 2009)

Mobiililaitteen komponentit vaikuttavat sen tehon kulutukseen. Esimerkiksi älypuhelimien yksittäisistä komponenteista eniten tehoa käyttävät laitteen näyttö ja sen valaiseminen sekä tietoliikenneyhteydet. Myös prosessorin kellotaajuudella on merkittävä vaikutus älypuhelimien tehon kulutukseen (Carroll & Heiser 2010). Prosessorien osalta mobiililaitteissa onkin parempi käyttää moniydinprosessoreja, koska näin saadaan vähennettyä yksittäisten prosessoriydinten kellotaajuutta ja laitteen prosessorien yhteenlaskettua akun käyttöä sekä ennen kaikkea akun lämpenemistä (van Berkel 2009). Useamman prosessorin hyödyntäminen vaatii kuitenkin monisäieohjelmointia.

Siirrettävyys vaikuttaa myös tietoturvaan. On esimerkiksi tärkeää, että mobiililaitetta ei jätetä valvomatta, kun sillä ollaan kirjautuneena luottamuksellisiin järjestelmiin. Mobiililaitteet kuitenkin unohtuvat esimerkiksi junaan työpöytäkoneita helpommin. Tästä syystä mobiililaitteilla luottamuksellisten tietojen on syytä olla salatussa muodossa. Vaihtoehtona on myös tietojen tallentaminen langattomien verkkojen avulla muualle kuin mobiililaitteeseen. Mahdollista on myös esimerkiksi tallentaa ei-luottamuksellinen osa

tiedoista laitteeseen ja näin vähentää tiedonsiirron määrää palvelimen ja mobiililaitteen välillä. (Forman & Zahorjan 1994)

Satyanarayanan (1996) määrittelee mobiilille tietojenkäsittelylle ominaiset haasteet hieman eri termein kuin Forman ja Zahorjan (1994), joskin molemmissa tutkimuksissa määrittelyjen yhteinen sisältö on melko lailla sama. Satyanarayananin mukaan haasteet ovat: mobiilien elementtien heikommalla resurssilla verrattuna staattisiin elementteihin, mobiiliuden ongelmaherkkyys, mobiilien yhteyden herkkyyden yhteydenlaadun heittelylle sekä yhteyden katkeamisille ja mobiilien laitteiden rajallinen energialähde. Molemmissa tutkimuksissa haasteiden määrittäminen mobiilille tietojenkäsittelylle ominaisiksi tarkoittaa, että ne ovat riippumattomia teknologian kehityksestä ja siitä syystä ne tulevat aina aiheuttamaan erityisvaatimuksensa mobiilien laitteiden ja sovellusten kehittäjille.

3.1.2 Sovelluskehitys mobiililaitteille

Mobiililaitteilla käytettävien sovellusten suunnittelu ja toteutus eroaa työpöytäsovellusten kehittämisestä esimerkiksi arkkitehtonisessa mielessä. Mobiililaitteet ovat muun muassa pienempiä, näytön tarkkuus on heikompi², laitteita käytetään esimerkiksi sormella tai kynällä näppäimistön ja hiiren sijaan, tietoliikenneyhteydet ovat virheherkemmät sekä eri laitteet voivat erota ominaisuuksiensa, kuten kokonsa, tallennuskapasiteettinsa tai suorituskykynsä, puolesta hyvinkin paljon. Mobiili ja työpöytälaitteiden eroavaisuuksista johtuen sovellusten suunnittelu niille on erilaista niin sovellusarkkitehtuurin kuin sovellusten käyttöliittymienkin osalta. Työpöytäkäyttöön suunniteltu sovellus ei useinkaan ole käytettävyyden tai tarpeenmukaisuuden suhteen optimaalinen mobiilikäyttöön eikä päinvastoin. (Oinas-Kukkonen & Kurkela, 2003)

Mobiiliuden erikoispiirteet ja rajoitteet on hyvä huomioida mobiilisovelluksen arkkitehtuuria suunnitellessa ja sovellusta toteuttaessa. Watsonin (1996) mukaan keskeistä mobiilisovellusten suunnittelun osalta on käsite hajautetusta sovelluksesta. Sovelluksen toiminnallisuuksia, kuten prosessointia ja tiedon tallennusta, on tällöin jaettu mobiililaitteen sekä palvelimen välille. Esimerkiksi laskennallisesti raskaampia operaatioita voidaan toteuttaa palvelimella, jolloin mobiililaitteella ainoastaan käynnistetään laskenta ja näytetään sen tulokset, millä voidaan esimerkiksi vähentää mobiililaitteen akun käyttöä. Hajautuksella voidaan tarkoittaa myös sovelluksen toiminnallisuuksien replikointia palvelimen ja sovelluksen välillä. Replikointia voidaan hyödyntää esimerkiksi tiedon tallentamiseen sekä laitteella että palvelimella. Hajautettu arkkitehtuuri on konseptiltaan samankaltainen kuin asiakas/palvelin-arkkitehtuuri, joskin siinä ero asiakkaan ja palvelimen välillä voi joskus hämärtyä johtuen esimerkiksi epävarmoista tietoliikenneyhteyksistä. Esimerkiksi yhteyden katketessa, mobiilisovellus voi joutua hoitamaan normaalisti palvelimella hoidettavia toiminnallisuuksia (Satyanarayanan 1996).

Hajautetussa arkkitehtuurissa osa toiminnallisuudesta on toteutettu mobiililaitteella ja osa palvelimella. Miten toiminnallisuudet näiden välillä on jaettu, riippuu esimerkiksi mobiililaitteen tallennuskapasiteetista, tehokkuudesta, yhteyden laadusta sekä tiedon määrästä ja muodosta. Esimerkiksi tallennuskapasiteetiltaan heikko mutta hyvät tietoliikenneyhteydet omaava mobiililaitte voidaan suunnitella hyödyntämään palvelinta

² Näytön tarkkuus saattaa jo tosin olla joillain mobiililaitteilla, kuten Applen iPad2:lla (2048*1536 pikseliä), samaa tasoa kuin työpöytätietokoneissa. Viitattu 21.8.2012. Saatavissa: <http://www.apple.com/ipad/features/>

enemmän kuin tallennuskapasiteetiltaan hyvä mutta huonot tietoliikenneyhteudet omaava mobiililaitte. Toiminnallisuuksien erottelu mobiililaitteen ja palvelimen välillä voi (Watson 1996) tai täytyy (Satyanarayanan 1996) olla myös dynaaminen. Tällöin riippuen esimerkiksi yhteyden laadusta tai käytön kontekstista, suurempi tai pienempi osa toiminnallisuudesta toteutetaan toisessa sovelluksen osassa. Toiminnallisuuksien hajautuksen dynaamisuus edellyttää, että mobiililaitte pystyy havainnoimaan ympäristönsä tilaa sekä reagoimaan tilan muutoksiin. (Watson 1996; Satyanarayanan 1996)

Toiminnallisuuksien hajautuksen dynaamisuus voidaan toteuttaa joko kokonaan yksittäisissä sovelluksissa tai kokonaan järjestelmän tasolla tai hyödyntäen molempia. Yksittäisten sovellusten tasolla toiminnallisuuksia hajauttaen, saadaan sovelluksen suorituskyky optimaaliseksi. Tällöin on kuitenkin ongelmana eri sovellusten toisistaan poikkeavien tarpeiden yhteensovittaminen. Yksi sovellus saattaisi esimerkiksi tällöin viedä kaikki järjestelmäresurssit. Kokonaan järjestelmän tasolla tehtävä hajautus taasen mahdollistaa resurssien tasaisemman jakamisen eri sovellusten välillä, joskin tällöin esimerkiksi osa suorituskyvystä menee systeemin hallintaan. Yleinen nykyisin käytössä oleva tapa on hyödyntää molempia tapoja. Tällöin saadaan osa molempien tapojen hyvistä puolista hyödynnettyä. Esimerkiksi nykyiset mobiililaitteiden käyttöjärjestelmät hallinnoivat laitteiden yksittäisiä komponentteja, kuten vaihtavat verkkoyhteyttä sen kuuluvuuden perusteella automaattisesti, kun taas yksittäiset sovellukset voivat hallita esimerkiksi sovelluskohtaisen tiedon tallennusta hajauttaen sitä yhteyden laadun perusteella palvelimen ja laitteen välillä. (Satyanarayanan 1996)

Mobiiliuden erityistarpeet voidaan huomioida myös esimerkiksi sovelluskehityksellä, joka mahdollistaa käyttäjille luontevasti tapahtuvan käyttötarpeiden toteuttamisen sekä sovelluskehittäjille kätevät työkalut sovellusten aikaansaamiseksi. Sovelluskehityksen onkin hyvä huomioida mobiilialustojen erityisominaisuudet, kuten laitteiden vaihteleva koko ja tietoliikenneyhteyksien vaihtelevat nopeudet sekä mahdollinen katkeaminen. Näin helpotetaan sekä kehittäjien työtä sovelluksen elinkaaren aikana että mahdollistetaan parempi sovelluksen käyttökokemus. Hyvä on myös arvioida, miten mobiilius voisi tehostaa tai helpottaa työntekoa sen sijaan, että vain yritetään mahdollistaa täysin samojen tehtävien hoitaminen sekä mobiili- että työpöytäsovelluksilla, joiden käyttö ja käytön konteksti ovat hyvin erilaiset. (Natchetoi ym. 2008)

Watson (1996) listaa joitain tekniikoita, joilla voidaan vähentää mobiililaitteen tietoliikenneyhteyksien virheherkkyyden tai heikon laadun vaikutusta sovelluksen käyttöön. Tekniikoita ovat: välimuisti, ennakoon lataaminen, tiedon koodaaminen, tarvepohjainen suoritus, futuurit sekä tiedon määrän vähentäminen. Tekniikoiden käyttäminen edellyttää kuitenkin tarkempaa tietoa sovelluksen käyttökohteesta, mistä syystä niiden automaattinen hyödyntäminen esimerkiksi järjestelmätasolla on vaikeaa. Tällöin ainoaksi ratkaisuksi tekniikoiden käyttöönnotolle jää sovelluskohtaisten ratkaisujen toteuttaminen.

Toteuttamalla sovellukseen tiedonhallinnan ominaisuuksia, on mahdollista vähentää tietoliikenneyhteyksien virheherkkyyden ja heikon laadun vaikutusta sovellukseen. Sovelluksella voidaan esimerkiksi tallentaa ennakoon mobiililaitteelle tulevassa käyttökontekstissa tarvittavia tietoja, mikäli laitteen tallennustila on riittävä. Myös käytön aikana voidaan hyödyntää mobiililaitteen lokaalia tallennustilaa. Tiedonhallinta mobiililaitteella vaatii kuitenkin ainakin osittaisen palvelimen tallennustoiminnallisuuksien replikoinnin mobiilisovellukseen sekä tietojen synkronoinnin toteuttamisen mobiililaitteen

ja palvelimen välille. Mahdollisesti tähän voidaan kuitenkin hyödyntää mobiilisovelluskehysjä tai -työkaluja, jotka osaavat toteuttaa tiedonhallinnan ominaisuuksia jossain määrin tai kokonaan automaattisesti.

Toiminnanohjausjärjestelmässä käyttöoikeuksien hallinta ja kontrollointi on tärkeä osa. Sopivilla käyttöoikeuksilla mahdollistetaan tarpeellisen tiedon ja prosessien saavutettavuus työntekijän näkökulmasta. Niillä voidaan myös estää yrityksen sisäisen tiedon saavutettavuus niin yrityksen sisällä kuin sen ulkopuolellakin henkilöiltä, joilla ei katsota olevan tarvetta tai oikeutta tiedon tarkastelemiseen. Käyttöoikeuksien hallinta on virheherkkyyden kannalta kriittinen. Mikäli käyttöoikeuksia ei ole rajattu riittävällä tasolla, on riskinä, että esimerkiksi kilpailijat pääsevät käsiksi yrityksen liiketoiminnalle kriittiseen tietoon. Virheiden välttämiseksi onkin hyvä, jos jokin muu osa kuin mobiilisovellus, kuten sovelluskehys tai palvelin, osaa huolehtia käyttöoikeuksista, jolloin vastuu tietoturvasta ei ole mobiilisovelluksen kehittäjällä, vaan vastuu on siirretty tietoturvan kannalta usein paremmin testatulle osalle sovelluksen kokonaisarkkitehtuuria. (Walter ym. 2004)

3.1.3 Mobiilikäyttökonteksti

Liikkuvan työnteon myötä työympäristö ja työnteon konteksti on tullut aikaisempaa monipuolisemmaksi. Työtä tehdään oman työpisteen – jos sellaista ylipäänsä edes on – lisäksi esimerkiksi kotona, liikennevälineissä, kahviloissa ja hotelleissa. Eri paikoissa, eri aikoina ja eri tilanteissa käytön konteksti on erilainen. Esimerkiksi ympärillä voi olla täysin eri ihmiset ja häiriötekijät, kuten melutaso, vaihtelevat. Eri konteksteissa myös tarpeet ovat usein erilaiset (Schilit ym. 1994). Esimerkiksi omalla työpisteellä on usein paremmin aikaa paneutua tarkkoihin tilastoihin edeltävän kuukauden myynnistä asiakkaalle, kun taas tämän luona vieraillessa on kokonaiskuvan saamiseksi tärkeää vain nähdä nopeasti muutama kuvaava luku edeltävän kuukauden myynnistä. Kontekstin automaattisella hyödyntämisellä onkin potentiaalia helpottaa työn tekoa yhä monimutkaisemmaksi muuttuvassa työympäristössä.

Schilit, Adams ja Want (1994) luokittelevat kontekstitietoiset sovellutukset neljään luokkaan, jotka ovat: läheisyysvalinta (proximate selection), automaattinen kontekstuaalinen uudelleen konfigurointi (automatic contextual reconfiguration), kontekstuaalinen informaatio ja komennot (contextual information and commands) sekä kontekstireaktiiviset toiminnot (context-triggered actions). Kaikkien luokkien sovelluksille on yhteistä kyky seurata ympäristöä ja reagoida siihen. Reagointitapa eroaa kuitenkin luokkien välillä.

Läheisyysvalinnassa käyttäjän sijaintia lähellä olevia kohteita korostetaan käyttöliittymässä. Korostaminen voi tapahtua esimerkiksi fontin kokoa muuttamalla tai järjestämällä listan kohteista sen perusteella, millä etäisyydellä kohteet ovat käyttäjästä. Läheisyysvalinnalla voidaan myös valita automaattisesti kohteita sijainnin perusteella. Esimerkiksi huoneesta toiseen siirryttäessä läheisyysvalintaa hyödyntävä sovellus voi tulostettaessa ehdottaa tulostinta huoneesta, johon siirrytään. Mahdollista on myös listata sijainnin perusteella lähistöllä olevia kohteita, kuten ravintoloita, ja päivittää listaa sen mukaan, miten sovelluksen käyttäjä liikkuu. (Schilit ym. 1994)

Automaattisella kontekstuaalisella uudelleen konfiguroinnilla voidaan lisätä uusia komponentteja, poistaa vanhoja tai muuttaa yhteyksiä komponenttien välillä riippuen kontekstista. Komponentit voivat olla esimerkiksi palvelimia, muita laitteita ja ohjelmia.

Sijainti on ilmeisin lähtökohta kontekstin hyödyntämiseen (Abowd ym. 1997), mutta kontekstia voidaan hyödyntää myös esimerkiksi ottamalla läsnä olevat ihmiset tai tila, jossa kullakin ajanhetkellä ollaan, mukaan kontekstiin. Schilit, Adams ja Want esimerkiksi loivat automaattisen kontekstuaalisen uudelleen konfiguroinnin luokkaan sopivan systeemin, jossa eri huoneissa on virtuaaliset piirtopöydät, joihin ihmiset pystyvät liittymään kämmentietokoneillaan huoneeseen tullessaan. (Schilit ym. 1994)

Ihmisten toiminta tietyissä tilanteissa on usein ennustettavissa. Kontekstuaalinen informaatio ja komennot –luokan sovellukset pyrkivätkin ottamaan tämän huomioon. Esimerkiksi ravintoloita haettaessa hakukone voi automaattisesti hyödyntää käyttäjän sijaintia ja tarjota lähistöllä sijaitsevia ravintoloita. Sovelluksen käyttöliittymä voi myös muuttua kontekstin perusteella. Esimerkiksi ruokaohjesovellus voisi kaupassa tarjota tuotteiden hinta- ja saatavuustietoja, kun taas keittiössä ollessa se voisi korostaa ruokaohjetta ja tarjota lisäohjeita ruoanlaiton eri työvaiheista. (Schilit ym. 1994)

Kontekstireaktiiviset toiminnot -luokka on samankaltainen kuin kontekstuaalinen informaatio ja komennot –luokka. Erona on kuitenkin, että edellisessä toiminnot suoritetaan automaattisesti kontekstin perusteella. Esimerkki tämän luokan sovelluksesta on Androidille kehitetty Toggle Settings³, jolla on mahdollista luoda sääntöjä, jotka muuttavat puhelimen asetuksia automaattisesti riippuen esimerkiksi sijainnista tai kellonajasta. (Schilit ym. 1994)

Kontekstin, liittyy se sitten sijaintiin tai ympäristöön, hyödyntämisellä sovelluksissa on valtavasti potentiaalia ja sen mahdollisista sovellutuksista on nähty varmasti vasta pieni osa. Pitkälle on jo tultu Schilitin, Adamsin ja Wantin automaattisesti konfiguroituvista virtuaalisista kirjoituspöydistä, kun nykyisin voidaan esimerkiksi sijainnin ja kuvantunnistuksen perusteella lisätä tietoa ympäristöstä videokameran kuvaan reaaliaikaisesti. Esimerkki tällaisesta lisätyn todellisuuden sovelluksesta on Layar⁴. Saatavilla olevan tietomäärän yhä kasvaessa ja tietoa keräävien antureiden kehittyessä, korostuu Abowd'n ym. (1997) esittämä ajatus, että kontekstin hyödyntämisessä on parempi, mitä enemmän tiedon keräämisestä ja hyödyntämisestä tapahtuu automaattisesti ilman käyttäjän syötettä. Tällöin käyttäjä voi paremmin keskittyä sen hetkiseen tehtäväänsä ja saa tukea sen suorittamiseen.

3.2 Mobiilisovellustyypit

Mobiilisovellukset voidaan jakaa neljään päätyyppiin: Mobiili web-sivu, natiivisovellus, web-sovellus ja hybridisovellus. Mobiili web-sivu on muuten vastaava kuin tavallinen www-sivu, mutta se on koon puolesta sopeutettu mobiililaitteelle sopivaksi. Web-sovellus käyttää mobiilin web-sivun tavoin tavallisia www-tekniikoita, mutta se on toteutettu vastaamaan ulkoasultaan ja käytettävyydeltään natiivia sovellusta. Natiivisovellus käyttää laitekohtaisia kirjastoja ja se täytyy toteuttaa jokaiselle laitteelle erikseen. Hybridisovellus voidaan kehittää ei-laitekohtaisena, mutta se käännetään laitekohtaiseksi ja siinä on mahdollista hyödyntää laitekohtaisia kirjastoja, joskin tämä saattaa edellyttää laitekohtaisen ohjelmakoodin kirjoittamista. (Faulk 2012)

³ Toggle Settings –sovellus Androidille. Viitattu 10.8.2012. Saatavissa: <http://www.togglesettings.com/>

⁴ Layar. Lisätyn todellisuuden sovellus iPhoneille ja Androidille. Viitattu 21.8.2012. Saatavissa: <http://www.layar.com/>

Mobiileille laitteille kehittämisessä haasteena on eri selainten ja päätelaitteiden välisten erojen huomioinen. Eri selaimet esimerkiksi saattavat asemoivat tekstiä eri tavalla, jolloin monesti web-sivua tai sovellusta tarvitsee testata useilla eri laitteilla kehityksen aikana. Päätelaitteet ovat usein myös erikokoisia ja erimuotoisia, jolloin esimerkiksi tekstimuotoista aineistoa kuvineen voi olla vaikea asemoida. Laitteita myös käytetään monilla eri tavoilla, kuten näppäimistöllä ja koskettamalla. Mobiili web-sivulla ja mobiili web-sovelluksilla eri alustojen huomioimisen tarve korostuu, kun taas natiivi- ja hybridisovelluksilla on jossain määrin yhtenäisempi laite- ja selainkanta. Kaikissa tapauksissa on kuitenkin tärkeää huomioida eri sovellusalustojen väliset erot ja sovellustarpeesta riippuen pyrkiä huomioimaan ne kehityksen aikana.

3.2.1 Mobiili-web-sivu

Mobiili web-sivu sopii tavallisten web-sivujen korvikkeeksi mobiileille käyttäjille. Mobiili web-sivu voi olla tekstimuotoinen ja se voi sisältää myös esimerkiksi kuvia. Se on helposti saavutettavissa ilman erillisten sovellusten lataamista ja sen avulla on mahdollista saada nopeasti tarvittava tieto hetkellisiin tarpeisiin, kuten osoitteen hakemiseen. Mobiili web-sivu ei kuitenkaan sovellu aina yritysympäristöön, jossa sovellustarpeet ovat usein staattisia sivulatauksia monimutkaisemmat.

Mobiili web-sivu on hyvä ratkaisu esimerkiksi pienten yritysten kotisivujen tekemiseen, jolloin esimerkiksi yksi tärkeimmistä asioista on, että kotisivuilla on yrityksen yhteystiedot selkeässä muodossa. Mobiili web-sivu on tällaiseen tarpeeseen usein halvempi ja yksinkertaisempi toteuttaa kuin esimerkiksi natiivisovellukset eri laitteille. Mobiili tekstimuotoinen web-sivu on myös helposti löydettävissä hakukoneiden avulla, kun taas natiivisovellus täytyisi ennen käyttöä löytää ja ladata laitteelle.

3.2.2 Natiivisovellus

Natiivisovelluksella tarkoitetaan laitekohtaisia kirjastoja suoraan käyttävää sovellusta, jonka ajaminen muilla laitteilla on vaikeaa tai mahdotonta. Natiivisovelluksia ovat esimerkiksi Applen iOS:lle tai Googlen Androidille tehdyt sovellukset. Natiivisovelluksella mahdollisten päätelaitteiden määrä on monesti vähäisempi kuin muilla sovellustyypeillä. Tämä helpottaa sovelluksen käyttöliittymän toteuttamista, koska vähäisempi päätelaitteiden määrä tarkoittaa vähäisempää määrää eroja eri laitteiden välillä.

Natiivisovellus soveltuu esimerkiksi sisällöltään tarkasti kohdistettujen tai käyttäjämääriltään suurten sovellustarpeiden täyttämiseen. Natiivisovellus voidaan tehdä esimerkiksi päivälehden digitaalisen version lukemiseen⁵.

3.2.3 Web-sovellus

Web-sovellus soveltuu usein web-sivua paremmin interaktiivisten tai reaaliaikaisesti päivittyvien sivustojen toteuttamiseen, vaikkakin käytetyt tekniikat ovat molemmissa tapauksissa samoja. Web-sovelluksella pyritään tarjoamaan natiivisovelluksia vastaava käyttökokemus, mikä on usein hankalaa tavallisilla web-sivuilla johtuen ennen kaikkea mobiililaitteiden pienestä koosta ja niiden käytön sekä käyttökontekstin erilaisuudesta verrattuna web-sivujen työpöytäkäyttöön. Web-sovelluksen etuna natiivisovellukseen

⁵ HS Digilehden iPad-versio. Viitattu 7.8.2012. Saatavissa: <http://asiakaspalvelu.hs.fi/tuoteperhe/hsdigi/ipadilla.html>

verrattuna on, että sovellusta ei tarvitse erikseen ladata, vaan se ”käynnistyy” automaattisesti. Haittapuolena on, että monet web-sovellusten interaktiivisemmat toiminnallisuudet, kuten kuvaajat, eivät välttämättä toimi eri laitteilla ja selaimilla samalla tavalla. Tämä aiheuttaa lisäkustannuksia kehitykseen, kun joudutaan huomioidaan eri alustojen erityistarpeet esimerkiksi tekemällä alustakohtaista ohjelmakoodia. Toinen vaihtoehto on käyttää jotain valmista kirjastoa sovellustarpeen⁶ toteuttamiseen, jolloin voidaan vähentää alustakohtaisen ohjelmakoodin tarvetta.

Nykyiset mobiilit web-sovellukset toteutetaan usein HTML5-spesifikaatioluonnoksiin kuuluvilla tekniikoilla, kuten sijainti (W3C 2012c) sekä lokaalitalennus (mm. W3C 2011). HTML5 (W3C 2012a) itsessään ei ole varsinainen standardi, vaan se kattaa määrittelystä riippuen vaihtelevan määrän eri web-tekniikoita. Osittain sen voi nähdä myös mainosterminä, jolla vakuutetaan tietyn tuotteen laatua mainostamalla sitä HTML5-yhteensopivana. Joka tapauksessa HTML5:n voidaan katsoa kattavan useat viime vuosina tulleet ja lähitulevaisuudessa yleistyvät web-tekniikat, jotka muokkaavat web-käyttöä yhä enemmän työpöytäkäytön suuntaan mahdollistavan monia työpöytäkäytössä totuttuja toiminnallisuuksia.

3.2.4 Hybridisovellus

Hybridisovellus kehitetään pääasiassa yhdestä ja samasta lähdekoodista, josta se käännetään eri alustoille natiiveiksi ohjelmiksi. Hybridisovellusten etuna on, että monissa tapauksissa, säästytään alustakohtaisen ohjelmakoodin tekemiseltä. Tämä vähentää sovelluskehitykseen tarvittavia resursseja. Toisaalta, koska eri alustat mahdollistavat eri toiminnallisuuksia, ei täysin samaa koodia voida käyttää eri alustoilla kaikissa tapauksissa, vaan tarvitaan sovellusspesifejä ohjelmakoodiosia, mikä taas vie sovelluksen kehitystä lähemmäs natiivien sovellusten kehitystä. Haittapuoleet ovat samat kuin natiivisovelluksilla. Esimerkiksi sovellus täytyy ennen käyttöä ladata. Hybridisovellusten käyttökohteet ovat hyvin samanlaiset kuin natiivisovelluksilla.

3.3 Mobiili paikkatietosovellus

Mobiilipaikkatietotekniikoita voidaan hyödyntää sekä ihmisten arjessa että liikkuvien työntekijöiden erilaisissa tarpeissa. Näin mahdollistetaan sijaintiin liittyvän tiedon hyödyntäminen ajasta ja paikasta riippumatta. Mobiilin paikkatiedon tekniikat eroavat kuitenkin perinteisten paikkatietojärjestelmien tekniikoista johtuen esimerkiksi langattomien tietoliikenneverkkojen ja päätelaitteiden asettamista rajoitteista. Näin ollen mobiileilla paikkatietosovelluksilla tehtävät toimet voivat olla hyvin erilaisia perinteisillä paikkatietosovelluksilla tehtäviin toimiin verrattuna. Esimerkiksi laajamittainen manuaalinen tiedon syöttäminen ei ole useinkaan tarkoituksenmukaista mobiililaitteilla. Mobiilius tarjoaa myös kuitenkin tiedon keräämiselle ja käytölle uusia mahdollisuuksia esimerkiksi automaattisen sijainnin hyödyntämisen myötä.

Nykyisten sekä mobiili- että työpöytäpaikkatietosovellusten ja -palveluiden keskivertokäyttäjä eroaa perinteisten työpöytäsovellusten tyypillisestä asiantuntijakäyttäjästä, jolla saattaa olla monivuotinen kokemus paikkatiedon parissa työskentelystä ja joka usein paikkatiedon käytön lisäksi myös tuottaa lisää aineistoa (Virrantaus ym. 2001).

⁶ Raphaël – JavaScript Library. Kirjasto selainriippumattomien vektorigrafiikoiden ja myös graafien tekoon. Viitattu 7.8.2012. Saatavissa: <http://raphaeljs.com/>

Paikkatiedon mobiilikäyttötarpeet ja -tapaukset eroavat perinteisistä stationäärisen ympäristön vastaavista. Mobiilikäytön tarpeet liittyvät enemmän juuri tiettyyn ajanhetkeen. Käyttäjät esimerkiksi tarvitsevat navigaattorilta tiedon käännöksestä juuri sopivalla hetkellä ennen kääntymistä. Mikäli ajo-opastus myöhästyy, on riskinä, että käyttäjä ajaa harhaan. Opastus täytyy myös tarjota käyttäjälle helposti ymmärrettävässä ja selkeässä muodossa sekä nopeasti, jotta käyttäjän ajaminen häiriintyy mahdollisimman vähän. Mobiilikäytössä syventävän informaation tarjoamista käyttäjälle suoraan on hyvä välttää, jotta varmistetaan tarpeellisen informaation välittyminen. Haasteena on kuitenkin turhan tai tarpeettoman sekä tarpeellisen informaation määrittelemisen ja millä tavoin varmistetaan, että tarvittava ja vain tarvittava informaatio välitetään käyttäjälle. Kontekstietoa, kuten sijaintia, hyödyntämällä voidaan mahdollistaa paremmin käyttäjän tarpeeseen sopivan informaation välittyminen, koska tietyssä kontekstissa käyttäjillä on usein tietyn tyyppiset tarpeet. (Reichenbacher 2001)

Perinteisten työpöytäpaikkatietosovellusten ja nykyisten usein web-pohjaisten paikkatietosovellusten käyttäjäryhmät eroavat asiantuntijuuden ja tiedon tuottamisen tasolla toisistaan. Web 2.0:n myötä on kuitenkin myös noviisikäyttäjille tullut mahdolliseksi uuden tiedon tuottaminen. He eivät enää siis ole pelkästään aineiston käyttäjiä. Imielinski ja Badrinath (1994) ennustivat tämän suunnanmuutoksen mobiileissa informaatiopalveluissa ja se on selvästi olemassa olevia sovelluksia tarkastellessa käynyt toteen. Esimerkiksi OpenStreetMap-palvelun⁷ aineisto on suurelta osin yksittäisten ihmisten omalla navigaattorillaan keräämää ja palveluun syöttämää eli aineiston keräys on toteutettu joukkoistamisen menetelmällä. Joukkoistamisessa kuilua asiantuntijoiden ja ei-asiantuntijoiden käyttäjien välillä pienennetään teknologian avulla, jotta saadaan suuremmat ei-asiantuntijakäyttäjien massat hyödynnettyä aineiston keräämisessä. Tämä edellyttää kuitenkin, että aineiston kerääminen ja syöttäminen ei vaadi asiantuntijatasoa ammattitaitoa (Howe 2006). Esimerkiksi OpenStreetMap:n kehittäjät ovat helpottaneet aineiston keräämistä ja tallentamista automatisoimalla nämä toimet mahdollisimman pitkälle. Joukkoistamisessa on kuitenkin automatisoinnin ja käyttäjien asiantuntemattomuuden myötä riskinä, että aineisto ei ole laadultaan riittävän hyvää moniin käyttötarkoituksiin. Kuitenkin, kontekstia, kuten sijaintia ja käyttäjän tietoja, tekniikan avulla hyödyntämällä, on mahdollista helpottaa tällaisten aineistojen keräämistä ja käyttöä.

Mobiiliuden olemukseen kuuluu erottamattomasti sijainnin jatkuva vaihtuminen (Imielinski ja Badrinath 1994). Sijainnin muuttuessa myös ympäristö eli laajemmin käytön konteksti muuttuvat myös jatkuvasti. Työpäivän aikana työntekijä voi tehdä töitä muun muassa kotona, autossa, toimistolla, asiakkaan luona, lentokoneessa ja kahvilassa. Eri paikoissa on eri kontekstit ja eri tarpeet. Esimerkiksi mobiilien paikkatietotarpeiden näkökulmasta autossa työntekijä voi tarvita reitin asiakkaan luo, kun taas kahvilassa hän voi olla kiinnostunut tietämään seuraavan asiakkaan osoitteen. Voisi olla työtä helpottavaa, jos mukana kulkevan mobiililaitteen sovellukset osaisivat tarjota työntekijälle sen hetkiseen tarpeeseen relevantin tiedon esimerkiksi automaattista kontekstuaalista informaatiota hyödyntämällä.

Reichenbacher (2001) ehdottaa neljää erilaista mobiilin ympäristön käyttäjän toimenpidettä tai käyttötapausta, jotka ovat: paikantaminen (locators), läheisyys (proximity), navigointi

⁷ OpenStreetMap – Avoin ja vapaasti muokattava maailmankartta. Viitattu 22.8.2012. Saatavissa: <http://www.openstreetmap.org/>

(navigation) ja tapahtumat (events). Paikantamisessa selvitetään, missä joku tai jokin on tai vaihtoehtoisesti mitä jossakin on. Sen tuloksena saadaan siis esimerkiksi maantieteelliset koordinaatit tai rakennuksen nimi selville. Läheisyydellä tarkoitetaan jonkin paikan ympäristöön liittyviä kyselyitä. Esimerkiksi läheisyyskäyttötapauksena voi olla kysely tietyn sijainnin ympäristön potentiaalisista asiakkaista. Kolmas luokka on navigointi, joka esimerkiksi vastaa kysymykseen, mikä on nopein reitti työpaikalta asiakkaan luo. Tapahtumat-luokan tapaukset vastaavat kysymykseen, mitä jossakin tapahtuu. Niiden tuloksena saadaan esimerkiksi selville liikenneneruuhkat tai katkokset kuljetusketjussa.

3.4 Toiminnanohjausjärjestelmä

Toiminnanohjausjärjestelmä (ERP) on yrityksen tietojärjestelmä, jonka tarkoituksena on tehostaa yrityksen toimintaa virtaviivaistamalla prosesseja informaation ja työn kannalta (Al-Mashari 2003). Toiminnanohjausjärjestelmätoimittajia ovat esimerkiksi SAP, Oracle ja Microsoft. Yrityksen tietojärjestelmän ytimessä on laaja tietokanta, jonne tietoa tallennetaan ja josta sitä haetaan liiketoimintatarpeiden täyttämiseen (Davenport 1998). Yrityksen toiminnanohjausjärjestelmään pyritään kokoamaan yrityksen eri osastojen tietotekniset toiminnot, jotta järjestelmä voisi keskitetysti palvella eri osastojen tarpeita (Botta-Genoullez & Millet 2005). Yhdistelemällä yrityksen tietojärjestelmätarpeita on mahdollista karsia päällekkäisiä toimintoja yrityksen eri osien välillä. Keskitetyn tietojärjestelmän avulla on myös mahdollista helpottaa muutoksen hallintaa.

Varsinaisia toiminnanohjausjärjestelmiä edeltävien yritysjärjestelmien historian voidaan katsoa alkavan 1960-luvulta, jolloin ensimmäiset materiaalisuunnittelujärjestelmät (MPR) levisivät. Nämä järjestelmät kattoivat lähinnä tuotantoon liittyvien materiaaliwaatimusten hallinnan. 1970- ja 1980-lukujen aikana nämä järjestelmät, joista alettiin käyttää nimitystä tuotannonhallintajärjestelmät (MPR II), kehittyivät kattamaan laajemman osan tuotannollisista prosesseista ja niihin alettiin integroida myös muita tuotannon kannalta olennaisia liiketoimintaprosesseja, kuten ostoa ja myyntiä. 1980-luvun lopulta 1990-luvulle siirryttäessä järjestelmien välisen integraation määrä yhä vain syveni ja tietojärjestelmäkokonaisuudet, joista alettiin käyttää nimitystä toiminnanohjausjärjestelmä (ERP), laajenivat tarjoamaan ratkaisuja kattavasti myös yritysten tuotannon ulkopuolisiin tietoteknisiin tarpeisiin, kuten henkilöstöhallintoon. (Jacobs & Weston 2006)

Toiminnanohjausjärjestelmä koostuu useasta osasta tai moduulista, kuten henkilöstön hallinta, myynti, tuotanto ja toimitusketjunhallinta. Jokainen moduuli on itsenäisesti konfiguroitavissa ja moduulit ovat yhteydessä toisiinsa standardoitujen rajapintojen välityksellä. Toiminnanohjausjärjestelmä kasaakin yrityksen sisäiset tietojärjestelmätarpeet paremmin hallittavaksi kokonaisuudeksi. Yrityksessä on myös ulkoisia tietojärjestelmiä, kuten asiakaspalvelun järjestelmät, jotka usein ovat myös yhteydessä toiminnanohjausjärjestelmään hyödyntäen tai täydentäen sen tietoja. (Aberdeen Group 2000)

Jako toiminnallisuuksien ja tarpeiden mukaan ulkoisiin ja sisäisiin yritysjärjestelmiin ei ole tarkka, mutta se helpottaa yrityksen tietojärjestelmäarkkitehtuurin kehittämistä konseptuaalisessa mielessä. Ulospäin suunnatuissa järjestelmissä ja sovelluksissa tärkeää on esimerkiksi hyvä käytettävyys ja siisti ulkoasu, kun taas sisäisen käytön järjestelmissä ei sovellusten ulkoasulle useinkaan aseteta yhtä tarkkoja vaatimuksia. Sisäisille järjestelmille ulkoasua olennaisempaa on kehittää yrityksen toimintaa esimerkiksi prosessien tehostamisen kautta. (Aberdeen Group 2000)

Toiminnanohjausjärjestelmän käyttöönotto on yritykselle iso operaatio johtuen toiminnanohjausjärjestelmien monimutkaisuudesta. Yrityksen täytyy valita tarvittavat moduulit eli implementoitavat toiminnallisuudet liiketoiminnan ja sisäisten prosessien tukemiseksi. Tämä vaatii kuitenkin syvällistä osaamista käyttöönotettavasta toiminnanohjausjärjestelmästä, mitä ei useinkaan löydy yrityksen sisältä. Toisaalta taas harvalla järjestelmätoimittajalla tai ulkoisella konsultilla on riittävä ymmärrys yrityksen liiketoiminnasta toiminnanohjausjärjestelmän tuntemuksen lisäksi. Tällöin seurauksena voi olla epäsuhta yrityksen liiketoimintavaatimusten sekä järjestelmän toiminnallisuuksien välillä. Toiminnanohjausjärjestelmää käyttöönotettaessa yrityksen on oltava valmis myös muuttamaan liiketoimintansa malleja, koska toiminnanohjausjärjestelmät ovat aina jollain tasolla yleisiä ja yleistettyjä ratkaisuja yrityksen spesifeihin tarpeisiin. Toiminnanohjausjärjestelmän käyttöönottoa ei voikaan nähdä yksittäisenä operaationa vaan jatkuvana prosessina, jossa järjestelmää muokataan vastaamaan paremmin ja paremmin liiketoiminnan muuttuvia tarpeita. (Davenport, 1998; Soh ym. 2000)

Toiminnanohjausjärjestelmä integroi eri prosesseja ja tietoja, joita kaikkia ei pystytä järjestelmää käyttöönottaessa määrittelemään. Tämän seurauksena on tärkeää, että tiedonvälitys tapahtuu tarkasti määriteltyjen tiedonvälityskerrosten kautta, jotta uusien tarpeiden ilmaantuessa on mahdollista hyödyntää jo kerättyä tietoa tai olemassa olevia prosesseja uusia tarpeita täyttäviä toiminnallisuuksia toteuttaessa ilman, että olemassa olevia prosesseja tarvitsisi muuttaa. Tämä on tärkeää sekä yrityksen sisäisten että ulkoisten järjestelmien osalta. Yksi keino mahdollistaa palveluiden uudelleenkäyttäminen, on noudattaa palvelukeskeisen arkkitehtuurin periaatteita, jossa esimerkiksi web-palvelut kirjataan ja kuvataan keskitettyyn palvelurekisteriin, josta niitä on mahdollista etsiä ja ottaa käyttöön uusien tarpeiden ilmaantuessa.

Yrityksen tietojärjestelmien välisestä integraatiosta johtuen tietoturvaan ja tarkoituksenmukaisiin järjestelmän käyttäjien käyttöoikeuksiin on syytä kiinnittää huomiota, koska elektronisen liiketoiminnan riskit liittyvät ennen kaikkea integroitavien järjestelmien sisältämään tietoon ja sen turvalliseen hallintaan (Chen ym. 2006). Esimerkiksi yhteydet yrityksen ulkoisista järjestelmistä sisäisiin on syytä suojata huolella, jotta pienennetään riskiä väärinkäytöksistä, kuten liikesalaisuuksien vuotamisesta kilpailijoiden käsiin. Toiminnanohjausjärjestelmissä tietoturvan tarve vielä korostuu, koska järjestelmän eri osat ovat tiukasti integroituneet toisiinsa. Tietoturvan tason kasvattaminen, vaikeuttaa kuitenkin työntekijöiden pääsyä työn kannalta relevanttiin tietoon, joten tietoturva on tasapainoilua suojan ja liiketoiminnan tehokkuuden välillä. Toiminnanohjausjärjestelmissä onkin eri tekniikoita, kuten kertakirjautuminen, tasapainon saavuttamisen helpottamiseksi.

3.4.1 SAP

SAP on SAP AG:n (perustettu 1972) toteuttama toiminnanohjausjärjestelmä, jota käyttää lähes 200 000 asiakasta (SAP 2012b) ja joka on suosituin toiminnanohjausjärjestelmäratkaisu erityisesti suurten yritysten keskuudessa (Panorama Consulting Group 2011). SAP tarjoaa toiminnanohjausjärjestelmäratkaisuja kuitenkin laaja-alaisesti mitä erilaisimmilla toimialoilla, kuten perusteollisuus ja tietoliikenneala, toimiville suurille ja myös pienille yrityksille.

SAP toiminnanohjausjärjestelmän kehitys voidaan jakaa neljään osaan. Ensimmäinen kehitysvaihe oli SAP R/2 (R tarkoittaa reaaliaikaista ja 2 kaksikerroksista), joka koostui

keskusyksiköllä olevasta tietokannasta ja sovelluslogiikkakerroksesta, joka hoiti reaaliaikaisen laskennan. 1992 SAP julkaisi SAP R/3, joka lisäsi SAP R/2:een verrattuna kolmannen esityskerroksen ja tarjosi asiakas-palvelin-mallin mukaisen toiminnanohjausjärjestelmäratkaisun. SAP R/3 oli myös jaettu moduuleihin, jotka kattoivat yrityksen liiketoiminnan tyypilliset toiminnot, kuten laskutuksen (FICO), henkilöstö- (HR) ja materiaalihallinnon (MM). Moduulit linkittyivät tarvittaessa toisiinsa, mutta ne oli myös mahdollista ottaa käyttöön erikseen. Kolmanneksi kehitysvaiheeksi SAP:ssa voidaan katsoa Internetin hyödyntäminen SAP Portal/mySAP:n kautta. Nämä avasivat SAP:n laajemman käytön myös yrityksen ulkopuolisille toimijoille, kuten asiakkaille extranet:n muodossa, kun ennen tätä SAP:n käyttö yrityksen ulkopuolisiin tehtäviin oli rajatumpaa. Uusin kehitysaalto SAP:ssa on mobiiliratkaisujen tarjoaminen Internetin välityksellä, mikä on merkittävä tekijä SAP:n tavoitteessa saavuttaa miljardin käyttäjän raja vuonna 2015(SAP 2012b). (Sanjeet ym. 2012)

SAP:lla on historiansa aikana ollut tapana tukeutua pitkälti talon sisällä kehitettyihin sovelluskehitysteknologioihin, joista esimerkkejä ovat ABAP-ohjelmintikieli ja WebDynpro-käyttöliittymäteknologia. Sen sijaan avoimen lähdekoodin teknologioiden käyttö on ollut vähäisempää. Jos niitä on käytetty, on niiden pääasiallisena tarkoituksena ollut mahdollistaa SAP:n integroiminen ulkopuolisiin järjestelmiin. SAP:n kuitenkin otti merkittävän askeleen avoimien teknologioiden suuntaan julkaistessaan vuosikymmen sitten Netweaver-alustan, jolle oli mahdollista kehittää web-sovelluksia Java-ohjelmointikielellä mahdollistaen näin SAP-kehityksen tuotteella, jota useammat sovelluskehittäjät osasivat. Siirtyminen avoimimpien teknologioiden suuntaan yhdistettynä tavoitteeseen saada yhä suurempi osa yritysten henkilöstöstä käyttämään SAP:ia edellyttää, että SAP:n on otettava yhä paremmin huomioon yrityksen ulkopuolisia teknologiakäyttäjiä, kuten web-käyttöliittymäohjeistuksia, sen sijaan, että kehitettäisiin aina oma ratkaisu yleiseen ongelmaan. (Moy 2011)

3.5 Mobiili yritysjärjestelmä

Mobiilien yritysjärjestelmien tavoitteena on mahdollistaa työnteke riippumatta työnteon paikasta. Tarkoituksena on, että työntekijät voivat hoitaa tarpeelliset tehtävät riippumatta siitä, ovat he sitten ensisijaisella työpisteellään tai jossain muualla. Esimerkiksi yrityksen mobiilisovelluksilla voidaan mahdollistaa tilauksen täyttäminen ja syöttäminen järjestelmään heti sen synnyttyä vaikkapa asiakkaan tiloissa. Näin tieto tilauksesta saadaan välitettyä esimerkiksi tuotantoon ilman työpaikalle siirtymisestä aiheutuvaa viivästymistä. Tilauksen saatuaan tuotanto voi heti aloittaa prosessin tuotteen valmistamiseksi asiakkaalle esimerkiksi tilaamalla, manuaalisesti tai automaattisesti, tarvittavat raaka-aineet. (Aberdeen Group 2000)

Mobiileissa yritysjärjestelmissä on esimerkiksi syytä huomioida tietoliikenneyhteyksien potentiaaliset häiriöt sekä vaihteleva kapasiteetti. Jos näitä ei huomioida, on riski, että mobiilisti tehty tilaus ei siirry ollenkaan yrityksen tietojärjestelmään, jolloin se voi jäädä kokonaan käsittelemättä. Myös tietoturvan rooli korostuu mobiileissa yritysjärjestelmissä. Esimerkiksi mikäli kadonnut puhelin päättyy kilpailijan käsiin, voi se aiheuttaa suunnatonta vahinkoa yrityksen liiketoiminnalle, jos mobiililaitteen ja -sovellusten tietoturvasta ei ole huolehdittu tällaisen riskitilanteen vaatimalla tarkkuudella. (Aberdeen Group 2000)

Mobilisoitujen yritysjärjestelmien tietoturvaa voidaan hallita mobiililaitteiden hallintajärjestelmille (MDM), jotka mahdollistavat esimerkiksi mobiililaitteiden ja -sovellusten etähallinnan. Hallintajärjestelmillä voidaan esimerkiksi lukita hävinnyt

puhelin, päivittää mobiilisovelluksia tai poistaa sovelluksen käyttöoikeus tietyiltä käyttäjiltä etänä.

Yrityksen tietoteknisiä tarpeita mobilisoidessa on syytä kiinnittää huomiota sijoitetun pääoman tuoton (ROI) lisäksi myös sovelluksen tarkoituksenmukaisuuteen, käytettävyyteen sekä teknisiin ominaisuuksiin, kuten sovelluksen arkkitehtuuriin sekä sovelluksen toteuttamisen vaatimiin muutoksiin yrityksen järjestelmissä. Muut edellä mainitut tarpeet voidaan nähdä jopa kokonaisuuden kannalta tärkeämpänä kuin sijoitetun pääoman tuotto (Aberdeen Group 2000).

Mobiilit yrityssovellukset voidaan jakaa neljään tyyppiin: prosessikeskeisiin sovelluksiin, tuottavuutta parantaviin sovelluksiin, kuluttajasovelluksiin ja analytiikkasovelluksiin. Prosessikeskeiset sovellukset ovat yrityksen toiminnan kannalta kriittisiä eli työntekijät tarvitsevat niitä saadakseen työnsä tehtyä. Tähän luokkaan kuuluvat esimerkiksi monet kenttätöissä tarvittavat sovellukset. Tuottavuutta parantavat sovellukset eivät ole kriittisiä työn tekemisen kannalta, mutta ne tehostavat työn tekemistä. Näin ollen niille voidaan monesti laskea suoraan sijoitetun pääoman tuotto (ROI). Tuottavuutta parantaville sovelluksille ominaista on tiedon hakemisen ja tehtävien hyväksymisen prosessien nopeuttaminen. Kuluttajasovellukset auttavat yrityksiä tavoittamaan kuluttajat. Kuluttajasovellukset eivät ole pelkästään markkinointisovelluksia, vaan ne tuovat liiketoiminnallista lisäarvoa loppukäyttäjien, eli tässä tapauksessa asiakkaiden, kautta esimerkiksi paremman saavutettavuuden tai tehostettujen prosessien kautta. Mobiileille analytiikkasovelluksille ominaista on mahdollistaa liikkuvassa työssä tarvittavan tiedon havainnollistaminen ja analysoiminen työntekijän tueksi. Analytiikkasovelluksilla voidaan esimerkiksi visualisoida kuvaajilla myyntiä asiakkaalle. (Sanjeet ym. 2012)

Koko yrityssovellusta ei usein voida mobilisoida tai se ei ole kannattavaa sovelluksen laajuudesta tai sen laskennan vaativuudesta johtuen. Yrityssovellus on myös usein yhteydessä useisiin muihin järjestelmiin ja näiden yhteyksien toimivuus voi olla liiketoiminnan kannalta kriittistä. Yrityssovelluksia mobilisoidessa voidaankin lähteä eri näkökulmasta. Jos analysoidaan mobiiliuden erityistarpeita jo sovelluksen käyttökohdetta mietittäessä, saatetaan löytää uusia innovatiivisiakin tapoja parantaa liiketoimintaprosessia mobiiliuden kautta. (Natchetoi ym. 2008)

Arkkitehtuurisessa mielessä voidaan nähdä kolme eri skenaariota, miten taustajärjestelmän yrityssovellus mobilisoidaan. Mobilisoinnin kohteena oleva sovellus on transaktiopohjainen eli se ottaa vastaan syötettä, tekee tietyt toimenpiteet sekä palauttaa tulosteen. (Bevis ym. 2002)

Ensimmäisessä skenaariossa sovelluksen ympärille voidaan rakentaa kääre (wrapper), joka hoitaa tiedon välittämisen mobiilisovelluksen ja taustajärjestelmän sovelluksen välillä. Kääreen kehittäminen voi kuitenkin olla haastavaa johtuen mobiilisovellusten erikoistarpeista. Esimerkiksi jos taustajärjestelmän sovellus tarvitsee paljon eri syötteitä, voi mobiilin sovelluksen käyttäminen olla aikaa vievää, jos sillä edellytetään täytettävän tietoa kaikkiin syötekenttiin. (Bevis ym. 2002)

Toisessa skenaariossa taustajärjestelmän tietoa käytetään ja muokataan suoraan ilman, että hyödynnetään taustajärjestelmän olemassa olevaa sovellusta. Tässä tapauksessa voidaan esimerkiksi joitain syötteitä jättää täyttämättä mobiilisovelluksen avulla ja täydentää tiedot myöhemmin sopivassa vaiheessa prosessia. taustajärjestelmän sovelluksen

toiminnallisuutta joudutaan kuitenkin usein kopioimaan, mikä vaikeuttaa sovellusten jatkokehitystä. (Bevis ym. 2002)

Kolmannessa skenaariossa taustajärjestelmän sovellus hyödyntää jo valmiiksi avointa integraatiomallia eli jo sovellusta kehitettäessä on huomioitu mahdollisuus hyödyntää taustajärjestelmän sovellusta useiden eri asiakassovellusten avulla. Tämä skenaario on palvelukeskeisen arkkitehtuurin periaatteiden mukainen. (Bevis ym. 2002)

3.5.1 Mobiili SAP

SAP tarjoaa ja tukee useita rinnan käytettäviä tai vaihtoehtoisia menetelmiä ja työkaluja mobiilien yrityssovellusten toteuttamiseen, koska yhtä työkalua tai yhtä menetelmää ei ole tarkoituksenmukaista käyttää kaikkien mobiilisovellustarpeiden ratkaisemiseen. Mobiilisovellukset voidaan toteuttaa sekä web-sovelluksina että mobiililaitteelle asennettavina hybridi- tai natiivisovelluksina. Mobiilisovellukset voidaan ohjelmoida kokonaan omina sovelluksinaan tai vaihtoehtoisesti ne voidaan toteuttaa hyödyntäen SAP:n mobiilialustaa (SAP Mobile Platform). Myös yhteydet taustajärjestelmiin voidaan hoitaa sekä SAP:n mobiilialustan avulla esimerkiksi SAP Netweaver Gateway:n kautta Open Data Protocol:ia⁸ (OData) hyödyntämällä että ilman mobiilialustaa hyödyntäen muun muassa etäfunktiokutsuja⁹ (RFC) SAP:iin tai HTTP(S)-välitteisiä www-sovelluspalveluja (web service). (Sanjeet ym. 2012)

Mobiilisovelluksia on mahdollista toteuttaa web-sovelluksina, hybridisovelluksina sekä natiiveina sovelluksina. Hybridisovelluksille SAP tarjoaa ajoalustaksi Hybrid Web Container:n, joka mahdollistaa esimerkiksi turvallisen tiedon tallentamisen, sovellusten versioinnin sekä kertakirjautumisen (SSO) hyödyntämisen. SAP tukee tai tarjoaa useita sovelluskehitystyökaluvaihtoehtoja kaikkien näiden sovellustyyppien toteuttamiseen, koska se ei halua tukeutua vain joihinkin yksittäisiin teknologioihin. Seuraavassa on esitetty joitain SAP:n suosittelemista teknologiavaihtoehdoista sovellustyypeittäin.

Natiivisovellukset: iOS ja Objective C iPhone ja iPad –alustoille, Java Androidille ja BlackBerryille sekä C# Windows Mobilelle esimerkiksi hyödyntäen Sybase Mobile SDK:ta.

Hybridisovellukset: HTML5 ja Javascript hyödyntäen Adobe PhoneGap (Apache Cordova), Sencha Touch tai Appcelerator Titanium -sovelluskehityksiä.

Web-sovellukset: JQueryMobile, Sencha Touch sekä User Interface Development Toolkit for HTML5 (SAP UI5).

Sovellustyyppit voidaan jakaa myös käyttöliittymän osalta kahteen luokkaan: natiivitekhnologioihin perustuviin ja www-tekhnologioihin perustuviin. Monet hybridisovellustekhnologiat, kuten Sencha Touch ja Adobe PhoneGap, hyödyntävät molempia teknologioita mahdollistaen joidenkin natiivien ominaisuuksien käytön natiivialustariippumattomilla www-tekniikoilla. (Sanjeet ym. 2012)

⁸ Open Data Protocol (OData) on avoin web-tekniikoihin perustuva rajapinta tiedon hakemiseen ja päivittämiseen. [Viitattu 27.11.2012]. Saatavissa: <http://www.odata.org/>.

⁹ Etäfunktiokutsu (Remote Function Call, RFC) on standardirajapinta SAP-järjestelmien väliseen kommunikointiin. [Viitattu 27.11.2012]. Saatavissa: http://help.sap.com/saphelp_nw04/helpdata/en/6f/1bd5b6a85b11d6b28500508b5d5211/content.htm.

SAP tarjoaa useasta eri tuotteesta koostuvan mobiilialustan yrityksen tietojärjestelmien mobilisoinnin tueksi. Mobilisointi on mahdollista myös ilman mobiilialustan hyödyntämistä, mutta silloin haasteena on hallita kasvavaa yrityssovelluskokonaisuutta sekä sovellusten käyttäjien että yrityksen tietohallinnon näkökulmasta. Mobiilialustan tuotteita ovat esimerkiksi SAP Netweaver Gateway, Afaria ja Sybase Unwired Platform (SUP). Lisäksi SAP tukee kolmannen osapuolen tuotteista tällä hetkellä ainakin Sencha Touch:ia, JQuery Mobilea, Adobe PhoneGap:ia, Appcelerator Titanium:ia. SAP tarjoaa myös Sybase Mobile SDK -kehitystyökalut natiivisovellusten luomiseen SAP SUP – alustalle. SAP Netweaver Gateway tarjoaa standardipohjaisia ratkaisuja mobiilisovellusten ja ulkopuolisten järjestelmien yhdistämiseksi keskenään tai SAP-järjestelmiin. Afaria on tarkoitettu mobiililaitteiden hallintaan (MDM) eli ratkaisemaan esimerkiksi yrityksen mobiiliuden tietoturvaan liittyviä ongelmia. SUP on alusta mobiilien yritysjärjestelmien toteuttamiseksi ja se tarjoaa välineet yrityksen tietojärjestelmien kokonaisvaltaiseen mobilisointiin tietoon yhdistämisen, sovellusten luomisen, sovellusten käyttämisen sekä yrityksen tiedon suojaamisen näkökulmasta. (Sanjeet ym. 2012)

SAP:n tarjoaman mobiilialustan tuotteiden tarkoituksena on tarjota yhteinen arkkitehtuuri, joka voi palvella kaikkia yrityksen osastoja, käyttäjiä ja liiketoimintaskenaarioita. Arkkitehtuurin on kuitenkin oltava tällöin joustava ainakin yksittäisten sovellusten näkökulmasta. SAP pyrkiikin mobiilialustallaan tarjoamaan yhdenmukaisia työkaluja enemmän yrityksen tietohallinnolle kuin yksittäisille käyttäjille tai sovelluskehittäjille. Mobiilialustan arkkitehtuuri mahdollistaakin esimerkiksi useiden eri sovelluskehitystyökalujen käytön, sekä erikseen että toistensa kanssa rinnan, sovelluskehityksessä. Mobiilialustan kaikkia komponentteja ei myöskään ole tarpeen ottaa käyttöön kerralla tai ylipäänsä ollenkaan. Esimerkiksi Netweaver Gateway Server:ia voidaan hyödyntää tiedon välittämiseen ilman SUP:n hallintatoiminnallisuuksia tai päinvastoin. Yhteinen arkkitehtuuri liittyykin enemmän yksittäisten sovellusten ajoympäristöön, jossa sovelluksia voidaan hallita keskitetysti esimerkiksi versioinnin, tietosuojan ja käyttöoikeuksien näkökulmasta. Mobiilialustan yhtenä tavoitteena on myös estää yksittäisten sovellusten siiloutumista tarjoamalla yhteiset toimintamallit mobiilien tarpeiden toteuttamiseen. Tavoitteena on esimerkiksi, että tiettyyn yksittäiseen tarpeeseen kehitettyjä sovelluksia ei ole useita yrityksen eri osastoilla. (Sanjeet ym. 2012; Sybase 2011)

Mobiilit sovellukset ovat usein tiettyyn yksittäiseen tarpeeseen kehitettyjä ja niiden käyttöikä on joskus lyhyt eikä se ole verrattavissa esimerkiksi yrityksen liiketoiminnan taustajärjestelmien käyttöikänsä. Tällaisiin tarpeisiin sovellusta kehitettäessä mahdollisella sovellusalustalla tai sovelluskehityksellä ja sen arkkitehtuurilla on suuri merkitys kehityksen kustannuksiin. Sovellusalustan pitäisi tarjota tarvittavat sovelluksen kehitystä ja elinkaarta tukevat toiminnallisuudet sopivassa tasapainossa. Sovellusalusta ei saisi kuitenkaan tehdä sovelluksen kehityksestä kalliimpaa ja sen tulisi mahdollistaa myös sovelluksen jatkokehittäminen. Sovellusalustan on myös hyvä olla riittävän kevyt, jotta sillä voidaan toteuttaa lyhyen käyttöiän sovelluksia kustannustehokkaasti. Toisaalta esimerkiksi sovellusalustan aiheuttamia kustannuksia ei kuitenkaan voida tarkastella pelkästään yksittäisten sovellusten kautta, koska sen tavoitteena on tukea ja helpottaa sovellusten hallintaa ja kehitystä, kun sovelluksia on useita. Muun muassa näistä syistä käytettävän sovellusalustan valitseminen ei ole yksinkertaista, vaan valinta riippuu sekä yrityksen tietohallinnon kokonaistarpeista että yksittäisten sovellusten tarpeista. (Sanjeet ym. 2012; Sybase 2011)

Tiedon hakemiseen taustajärjestelmistä SAP-mobiili-ympäristössä on monia vaihtoehtoja, kuten www-sovelluspalvelut SAP XI:n tai Netweaver Portal:n kautta, Simple Object Access Protocol - (SOAP) tai OData-välitteiset palvelut Netweaver Gateway:n kautta tai Mobile Business Object -perusteiset (MBO) palvelut SAP SUP:n avulla. Näistä Netweaver Gateway ja OData on SAP:n suosittelema vaihtoehto. ODataa käytettäessä kaikki sovelluksen tarvitsema tieto voidaan määritellä taustajärjestelmässä eikä Netweaver Gateway:lle tarvitse asentaa mitään. Näin helpotetaan sovelluksen kehitystä, käyttöönottoa sekä ylläpitoa. Toinen SAP:n suosittelema vaihtoehto mobiilisovellusten tietotarpeiden täyttämiseen on MBO, joka tukee sovelluksen tietotarpeita tarjoamalla välineet tiedon mallintamiseen ja mallin pohjalta sovelluskoodin luomiseen. MBO tukee myös tiedon synkronointia sekä tarjoaa kehityksen turvalliselle tiedon välitykselle. Sekä OData että MBO-pohjaiset ratkaisut on kehitetty helpottamaan yleisten mobiilissa sovelluskehityksessä tarvittavien sovellustoiminnallisuuksien ratkaisuja määrittelyllä ja konfiguroitavalla tavalla. (Katz 2012; Sanjeet 2012)

3.6 Paikkatieto yritysjärjestelmissä

Yritysjärjestelmien tavoitteena on tukea liiketoimintaa esimerkiksi tehostamalla prosesseja ja tukemalla päätöksentekoa. Yrityksen päätöksentekoprosessissa tavoitteet on usein selkeästi määriteltä. Paikkatietojärjestelmillä tehtävät analyysit eivät kuitenkaan itsessään välttämättä tue päätöksentekoa johtuen paikkatietoon liittyvien ongelmien monimutkaisuudesta ja niiden vaikeasta tavoitteisiin suhteutettavuudesta. Paikkatietoa päätöksenteon apuna hyödyntäessä onkin vaikea rakentaa päätöksentekoa tukeva malli, jota paikkatietoanalyysillä voitaisiin analysoida. (Densham 1991)

Spatiaaliset päätöksentekotukijärjestelmät (spatial decision support systems, SDSS) auttavat päätöksentekijöitä monimutkaisten paikkatietoon liittyvien ongelmien ratkaisussa tukemalla päätöksentekoprosessia koko sen aikana ongelma-alueen määrittelystä varsinaisen analyysin tekoon. Spatiaaliset päätöksentekotukijärjestelmät mahdollistavat erilaisten ratkaisujen kokeilemisen interaktiivisesti ja vaihtoehtoisia menetelmiä hyödyntäen. Ne ovat myös usein rajatumpaan käyttötarkoitukseen suunniteltuja kuin varsinaiset paikkatietojärjestelmät. (Densham 1991)

Paikkatietojärjestelmän ja toiminnanohjausjärjestelmät ovat usein erillisiä järjestelmiä. Paikkatietojärjestelmässä voidaan esimerkiksi säilyttää paikkaan liittyviä tietoja ja se voi tarjota näitä tietoja sekä tietoihin tehtäviä paikkatietoanalyysijä toiminnanohjausjärjestelmän käyttöön. Paikkatietojärjestelmä voi myös hyödyntää toiminnanohjausjärjestelmän tietoja analyysien teossa. Paikkatieto- ja toiminnanohjausjärjestelmien integraatio onkin olennainen osa spatiaalisia päätöksentekojärjestelmiä. (ESRI 2007)

Paikkatietojärjestelmien ja SAP-järjestelmien integraatio voidaan toteuttaa esimerkiksi www-sovelluspalveluiden avulla. Integraatio voidaan toteuttaa myös syvemmällä tasolla hyödyntämällä kolmannen osapuolen tuotteita, kuten GISConnex¹⁰. Tämä vähentää tarvetta täysin räätälöidylle järjestelmälle. Useissa käyttötapauksissa paikkatietojärjestelmän ja SAP:n yhdistäminen vaatii kuitenkin pitkälle räätälöidyn ratkaisun toteuttamista. (ESRI 2007)

¹⁰ GISConnex on SAP:ia ja paikkatietoa yhdistävä tuote yrityksen omaisuususerien hallintaan. [Viitattu 9.11.2012]. Saatavissa: <http://www.gisconnex.com/index.asp>.

4 Mobiilin SAP-paikkatietosovelluksen toiminnallisuudet

Mobiilisovellusten käytössä korostuu käytön kontekstin vaikutus. Sovelluksia käytetään eri tilanteissa, joissa käyttötarpeet ovat usein erilaiset. Toisistaan poikkeavat kontekstiriippuvaiset tarpeet tai sovellustoiminnallisuustarpeet eivät kuitenkaan tarkoita, etteikö useimmissa mobiilisovelluksissa voitaisi tunnistaa tiettyjä sovellusominaisuuksia ja sovellustoimintoja, joita useimmat sovellukset käyttävät tai joita niissä on. Esimerkiksi melkeinpä mobiilisovelluksesta riippumatta, tarvitsee taustajärjestelmistä hakea tai sinne tallentaa jotain tietoa. Nämä sovellusominaisuudet voidaan nähdä tärkeimpinä, koska ne toistuvat usein sovelluksen käyttökohteesta riippumatta. Monet tärkeimmistä sovellusominaisuuksista myös liittyvät mobiilin tietojenkäsittelyn ominaispiirteisiin, kuten vaihtuvaan käytön kontekstiin tai rajoittuneisiin tietoliikenneyhteyksiin.

Mobiileissa paikkatieto-SAP -sovelluksissa sekä paikkatiedon että SAP-yritysjärjestelmän puolelta on omia yleisiä sovellusominaisuuksia. Paikkatiedon osalta tällaisia ovat esimerkiksi kartta, paikannus, navigointi ja paikkatietoanalyysit. Mobiiliin SAP-sovellukseen liittyvät muun muassa yhdistettävyyden taustajärjestelmiin niin SAP:n kuin muidenkin järjestelmien näkökulmasta, käyttöoikeuksien hallinta ja tiedonhallinta mobiililaitteella. Alaluvuissa on kuvattu tärkeimmiksi arvioituja sovellusominaisuuksia tarkemmin.

Sovelluksen käyttökohde vaikuttaa myös osittain siihen, mitkä sovellusominaisuudet ovat tärkeimpiä. Esimerkiksi vapaa-ajan käyttöön mobiilipeliä toteutettaessa, ei sovelluksen täsmällisen tarkalla pisteiden laskentalogiikalla tai pisteiden taustajärjestelmään synkronoinnilla ole samoja tarkkuusvaatimuksia, kuin pankin rahasiirtojen apuna käytettävän sovelluksen rahasiirtojen synkronoinnin toimivuuden tai tietoturvan tason osalta on. Mobiilipelissä tärkeämpää voi esimerkiksi olla, että sovelluksen käyttöliittymä on toimiva.

4.1 Mobiilin paikkatiedon sovellustoiminnallisuudet

Tässä luvussa kuvataan tärkeimmiksi arvioituja paikkatietoon liittyviä mobiilisovellustoiminnallisuuksia. Tarkasteltavia mobiilisovellustoiminnallisuuksia ovat: kartta, paikannus, navigointi ja paikkatietoanalyysit.

Reichenbacherin (2001) esittämät mobiilin ympäristön käyttötapaukset: paikantaminen, läheisyys, navigointi ja tapahtumat, linkittyvät mobiilisovellustoiminnallisiin, joskin luokittelu on hieman erilainen. Paikantaminen sisältää paikannustoiminnallisuuden, mutta myös joitain paikkatietoanalyysejä, kuten vastakkaisen geokoodauksen (reverse geocoding), jolla muunnetaan suora sijainti epäsuoraksi. Läheisyys ja tapahtumat luokkien käyttötapaukset taas ovat paikkatietoanalyysejä, jotka tuottavat uutta tietoa yhdistelemällä olemassa olevaa tietoa. Navigointi-käyttötapausta vastaa navigointitoiminnallisuutta, jonka tosin voi katsoa olevan myös analyysitoiminnallisuutta.

4.1.1 Kartta

Kartat ovat säilyttäneet asemansa pääasiallisena spatiaalisen tiedon välityksen tapana myös mobiiliin paikkatiedon hyödyntämiseen siirryttäessä. Mobiilikarttoja voidaan käyttää esimerkiksi uusissa kaupungeissa suunniteltaessa, ajoneuvojen sijaintitietojen esittämisessä sekä kokonaiskuvan saamiseksi tietyn alueen kiinteistöjen hintatietojen

muodostumisesta. Hyvin suunniteltujen mobiilikarttojen avulla ihmiset, niin töissä kuin vapaa-ajallakin, voivat parantaa käsitystä ympäristöstään. (Meng ym. 2004)

Mobiilius tuo kuitenkin mukanaan uusia haasteita esimerkiksi kartografisessa mielessä. Mobiililaitteiden näyttö on yleensä pieni, joten kartalla voidaan esittää rajatun informaatia ja symboleita kuin isommalla työpöytätietokoneen näytöllä. Erilaisilla tekniikoilla, kuten kartografisella yleistämällä tai tilanteeseen ja ympäristöön mukautuvilla eli kontekstiadaptiivisilla sovelluksilla (Reichenbacher 2001; Reichenbacher 2004), voidaan jossain määrin helpottaa käyttäjälle aiheutuvaa kognitiivista kuormaa.

Mobiiliin käyttökontekstiin liittyy usein käyttöä vaikeuttavia tekijöitä. Esimerkiksi autoa ajaessa kartan raahaaminen mobiililaitteen kosketusnäytöstä sormella vetämällä on vaikeaa. Mobiilikartan käytön olisikin oltava käyttäjän sen hetkiseen toiminnalliseen tarpeeseen saumattomasti yhdistyvää ja käytön tulisi onnistua ilman, että se häiritsee käyttäjän sen hetkisen toiminnallisen tarpeen suorittamista. Yhtäältä vaatimus mobiilikartan häiritsemättömyydestä ja toisaalta mobiilikartan käyttäjän toiminnan helpottajana ovat keskenään ristiriitaisia, mikä asettaakin erityishaasteensa mobiilikarttojen ja ylipäänsä mobiilisovellusten kehittämiseen. (Meng ym. 2004)

4.1.2 Paikannus

Nykyiset mobiililaitteet pystyvät määrittämään sijaintinsa esimerkiksi GPS:n ja mobiiliverkon avulla. Myös erilaiset sisäpaikannustekniikat, kuten langattomiin verkkoihin, radiotaajuuden etätunnistukseen (RFID) tai infrapunavaloon perustuvat paikannustekniikat, kehittyvät jatkuvasti, vaikkakaan niiden osalta ei vielä ole GPS:ään verrattavaa yleisesti käytössä olevaa toteutusta olemassa. (Hui ym. 2007) Sijainnin määrittäminen mobiilikäytössä mahdollistaa sijaintiin perustuvien kontekstisidonnaisten palveluiden toteuttamisen. Paikannusmahdollisuuden voikin katsoa olevan lisäarvoa tuova väline liikkuvan työntekijän tarpeisiin.

Paikannukseen liittyy sijaintitiedon lisäksi myös laitteen orientaatio eli suuntima, johon laite osoittaa. Mobiililaitteissa orientaatioinformaatiota tuottavat digitaalinen kompassi ja gyrospooppi. Kompassi tuottaa tiedon ilmansuunnasta ja gyrospooppi laitteen asennosta suhteessa maahan. Orientaatiotiedon perusteella esimerkiksi kartta voidaan suunnata mobiililaitteen käyttäjän katselusuunnan mukaisesti. Orientaatiota hyödynnetään myös esimerkiksi lisätyn todellisuuden sovelluksissa, kuten Layar. (Reichenbacher 2001)

4.1.3 Navigointi

Navigointitarve liittyy olennaisesti uudessa paikassa liikkumiseen. Esimerkiksi myyjät voivat selvittää seuraavan asiakkaan sijainnin karttaa apuna käyttäen. Nykyisin kartan rinnalla tai sen sijaan on alettu käyttää mobiililaitteita navigoinnissa. GPS vastaanottimen sisältävillä matkapuhelimilla on mahdollista selvittää reitittämällä esimerkiksi lyhin, nopein tai helpoin reitti kohteeseen ja näin esimerkiksi auttaa myyjää seuraavan asiakkaan sijainnin löytämisessä. Reitin laskennassa hyödynnetään usein palvelimia, mutta reititys on mahdollista tehdä myös pelkästään laitteella. Käyttäjän sijaintia voidaan myös seurata säännöllisesti ohjelmistorajapintojen, kuten W3C:n (2012b) laiteorientaatio-spesifikaation (DeviceOrientation) määrittelemän rajapinnan avulla. Navigointiin sisältyy myös orientaatioon liittyvät kompassi-, kiihtyvyysanturi- ja gyrospooppi-toiminnallisuudet, joiden avulla käyttäjä voi suhteuttaa kulku- tai katselusuuntansa ympäristöönsä. Orientaatiotoiminnallisuuksia voidaan hyödyntää

esimerkiksi lisätyn todellisuuden sovelluksissa tai mahdollistamalla totutusta poikkeava käyttöliittymän ja sen toiminnallisuuden käyttö, kuten tietojen uudelleen päivittäminen laitetta ravistamalla. Myös esimerkiksi varastosovelluksessa hyllyn suuntaan mobiililaitteella osoittamalla voi olla mahdollista nähdä mobiililaitteen näytöllä visualisointi hyllyssä olevan nimekkeen varastosaldosta ja nimekkeen menekistä viime aikoina, kun hyödynnetään mobiililaitteen orientaatio-ominaisuuksia osana sovellusta.

Mobiililaitteen käyttäjä navigoi usein tilanteessa, jossa hänen täytyy nopeasti ja selkeästi saada selville reitti tai suunta. Tällöin visuaalinen palaute mobiililaitteen pienellä ruudulla ei välttämättä ole tehokkain tapa välittää informaatiota. Navigoinnissa hyödynnetäänkin usein esimerkiksi ääniä ja puhuttuja ohjeita tiedon välityksen apuna. Esimerkiksi navigointisovellus opastaa autolla ajavaa ääniohjeilla kääntymään tietyn matkan päästä tai ajan kuluttua. Navigoinnin apuna käytettävissä visualisoinneissa voidaan myös käyttää hyvin yksinkertaistettuja 2-ulotteisia tai 3-ulotteisia karttoja, joissa tiettyjä kohteita tai piirteitä on ympäristön hahmottamisen helpottamiseksi korostettu. (Kray ym. 2003)

4.1.4 Paikkatietoanalyysit

Paikkatietoanalyysien avulla voidaan luoda uutta tietoa ja saada ymmärrystä ympäristöstämme yhdistelemällä olemassa olevia tai simuloituja tietoa ja tietolähteitä. Paikkatietoanalyysissä tietoalkion spatiaalisilla ominaisuuksilla on erityismerkitys, sillä lähellä toisiaan olevilla tietoalkioilla niiden ominaisuudet korreloivat keskenään enemmän kuin kaukana toisistaan sijaitsevilla tietoalkioilla. Tästä ilmiöstä käytetään nimitystä spatiaalinen autokorrelaatio. Spatiaalisen autokorrelaation lisäksi myös sijainnilla on erityismerkitys paikkatietoanalyysissä. Käyttäjän sijainnista, joka mobiililaitteilla on luontevaa selvittää, riippuen hänellä on erilaiset tarpeet, jotka huomioimalla paikkatietoanalyysillä on mahdollista tuottaa käyttäjälle käyttökelpoisempaa informaatiota. Tiedon suuresta määrästä ja dynaamisuudesta sekä mobiililaitteen rajallisesta kapasiteetista johtuen tämä voi kuitenkin olla vaikeaa toteuttaa pelkästään mobiililaitteessa, vaan sen lisäksi tarvitaan usein myös palvelinta tai palvelimia, jolloin mobiililaitteen tehtävänä analyysissä on tuottaa kontekstisidonnainen syöte analyyseille. (Reichenbacher 2001)

Mobiililaitteilla tehtävien paikkatietoanalyysien tehtävänä on usein helpottaa käyttäjää sillä hetkellä ajankohtaisen yksinkertaisen ongelman ratkaisussa. Käyttäjän tarve liittyy usein olennaisesti hänen sen hetkiseen sijaintiinsa tai sen ympäristöön. Ongelma voi olla esimerkiksi missä mobiililaitteen käyttäjän lähistöllä on potentiaalisia asiakkaita. Tällöin voidaan tarvita esimerkiksi geokoodausta eli kyselyä osoitteen selvittämiseen mobiililaitteen koordinaattien perusteella. Vaihtoehtoisesti käyttäjä voi olla kiinnostunut selvittämään, missä jokin paikka, jonka osoite on tiedossa, on. Tällöin hänen tietämänsä osoitteen perusteella tarvitaan koordinaatit, joiden selvittämiseen voidaan käyttää vastakkaista geokoodausta. (Reichenbacher 2001)

Mobiilit paikkatietotapaukset voidaan jakaa kahteen luokkaan: läheisyys-tapaukset ja tapahtuma-tapaukset. Läheisyystapauksissa eli -analyysissä selvitetään esimerkiksi, missä ovat lähimmät tietyn tyyppiset tai tietyn ominaisuuden omaavat kohteet tai mitä kohteita on tietyn matkan etäisyydellä. Tapahtumatapauksilla taas selvitetään mitä tiettyssä paikassa tapahtuu tai missä tapahtuu tiettyä asiaa. Esimerkki tapahtuma-analyysistä on matkalla olevien liikennetukoksien kysely. Tämän tyyppiset kyselyt on yleensä toteutettu paikkatietoon erikoistuneilla palvelimilla. Nykyisten älypuhelisten suorituskyky kuitenkin

mahdollistaa monimutkaistenkin paikkatietoanalyysien toteuttamisen ilman palvelinta. Mobiilisovelluksilla suoraan tehtävien analyysien rajoitteena onkin ehkä eniten paikkatietoanalyysijä tarjoavien sovelluskirjastojen puute. (Reichenbacher 2001)

4.2 SAP:n mobiilisovellustoiminnallisuudet

Tässä luvussa kuvataan tärkeimmiksi arvioituja mobiilisovellustoiminnallisuuksia, jotka liittyvät toiminnanohjausjärjestelmiin ja näistä tarkemmin SAP:iin. Käsiteltäviä mobiilisovellustoiminnallisuuksia ovat: yhdistettävyyden SAP:n dataan, käyttöoikeuksien hallinta sekä tiedonhallinta mobiililaitteella. Varsinkin älypuhelimilla moniydinarkkitehtuuri ja moniajo ovat yhdet tärkeimmistä tekijöistä puhelinten suorituskyvyn ja sitä kautta käytettävyyden lisäämisen kannalta (van Berkel 2009). Niitä ei kuitenkaan arvioida yksittäisten sovellusten sisäisten toiminnallisuuksien osalta tärkeimpiin kuuluviksi, koska moniydinarkkitehtuuria voidaan hyödyntää puhelimissa esimerkiksi eri komponenttien hallintaan hyvin jo järjestelmätasolla ja lähinnä sovellusten sisäinen moniajo parantaisi suorituskykyä vain hyvin monimutkaisia sovelluksia ajettaessa.

4.2.1 Yhdistettävyyden SAP-dataan

Ehkä tärkeimpänä toiminnallisuutena voidaan nähdä yhdistettävyyden toiminnanohjausjärjestelmän, joista tässä keskitytään SAP:iin, tietoihin sekä tiedon haun että tiedon tallentamisen näkökulmasta. Yhdistettävyyden liittyy usein myös tarve tiedon salaukselle sitä siirtäessä. Tiedonsiirrossa voidaan esimerkiksi käyttää SSL-salausprotokollaa. Mobiililaitteen rajallisesta tallennuskapasiteetista tai tiedon muutosnopeudesta johtuen laitteeseen ei yleensä voi tai ei ole tarkoituksenmukaista tallentaa kaikkea tarvittavaa tietoa etukäteen. Mobiilit yrityssovellukset ovatkin usein yhteydessä taustajärjestelmään tai –järjestelmiin, koska tällä tavoin toimien, ei tarvitse ladata ennakoon tarvittavia tietoja ja aineistoja sovelluksen liikkeessä tapahtuvaa käyttöä varten.

Mobiilisovellusten yhdistämiseen SAP-dataan on useita vaihtoehtoja, kuten SAP:n etäsovelluskutsut (RFC) ja www-sovelluspalvelut. SAP on viime aikoina siirtynyt hyödyntämään määrittelyltään avointa OData-tiedonvälitystapaa enenevässä määrin mahdollistaakseen avoimemman kolmannen osapuolen järjestelmien integroimisen SAP-järjestelmiin. Yrityksen mobilisoinnin kannalta varteenotettava vaihtoehto on myös MBO-palvelut, jotka ovat osa SUP:ia. Monista tiedonsiirtovaihtoehtoista johtuen, valinta käytettävän tiedonsiirtomenetelmän osalta ei ole aina selkeää. Kaikki sovelluskehikset eivät myöskään mahdollista kaikkien eri tiedonsiirtomenetelmien käyttämistä, mikä on myös syytä huomioda yrityksen mobiilisovellusarkkitehtuuria suunnitellessa sekä sovelluskehiksyä valitessa. Esimerkiksi tukea tiedon työntämiselle palvelimelta mobiilisovelluksiin eivät kaikki sovelluskehikset tue.

Etäsovelluskutsut ovat SAP:n kehittämä suljettu rajapinta tiedon siirtämiseen SAP-järjestelmien tai SAP:n ja kolmannen osapuolen järjestelmien välillä. Niiden etuna on, että ne ovat olleet pitkään vakiintuneessa käytössä. Ongelmana on kuitenkin, että niiden turvallinen ja helppo mobiilikäyttö vaatii yleensä välityspalvelimen, kuten Netweaver Application Server, joka hoitaa varsinaiset etäsovelluskutsut sekä esimerkiksi käyttäjän autentikoinnin. Tämä edellyttää usein sovelluskehitystä välityspalvelimelle. Toisaalta tarve välityspalvelimelle ei lisää mobiilisovellusten kustannuksia palvelinarkkitehtuurin näkökulmasta, jos välityspalvelin on jo käytössä yrityksessä. Näin monesti onkin, koska

välityspalvelinta on voitu aiemmin hyödyntää jo esimerkiksi yrityksen portaalisovelluksissa.

OData sekä MBO on molemmat kehitetty mobiiliuden erityistarpeet huomioon ottaen. Nämä molemmat mahdollistavat mobiilisovelluksen tarvitseman datan konfiguroinnin taustajärjestelmässä, jolloin säästetään kehityskustannuksissa. Molempien tuotteiden käyttö vaatii kuitenkin maksullisten palvelimien käyttämisen. OData on kehitetty ratkaisuksi avoimempaan tiedonvälitykseen muun muassa mobiilien sovellustarpeiden osalta ja se vaatii SAP Netweaver Gateway:n. SAP Netweaver Gateway:llä palvelut konfiguroidaan esimerkiksi olemassa olevien RFC-rajapintojen päälle. Konfiguroinnilla mahdollistetaan erilaisten tarpeiden huomioon ottaminen ilman tarvetta muokata taustajärjestelmän rajapintaa. MBO on kehitetty erityisesti mobiilisovellusten tarpeisiin ja se toimii vastaavalla tavalla kuin OData, joskin se vaatii SUP-palvelimen, jossa mobiilisovelluksen tietotarpeet määritellään. Ongelmana OData:a tai SUP:ia käyttäessä on lisenssikustannusten mahdollinen nousu verrattuna tilanteeseen, jossa käytetään vain RFC-kutsuja olemassa olevien välityspalvelimien kautta. OData:n ja MBO:n suunnittelussa on kuitenkin otettu huomioon mobiilitarpeet ja ne mahdollistavat yrityksen kattavamman mobilisoinnin kuin esimerkiksi pelkkien RFC-kutsujen tai www-sovelluspalvelujen käyttö. (Sanjeet ym. 2012)

4.2.2 Käyttöoikeuksien hallinta

Käyttöoikeuksien hallinnan tarkoituksena on mahdollistaa työntekijöiden pääsy heidän työnsä kannalta tarpeellisiin tietoihin. Käyttöoikeuksien hallintaan kuuluu sekä käyttäjän tunnistaminen eli autentikointi että käyttäjän käyttöoikeuksien validoiminen eli autorisointi. Käyttöoikeuksien hallinnalla on saavutettavissa parempi tietoturvan taso.

Tietoturvallisuus on olennainen tekijä mobiililaitteilla. Mobiilisovelluksissa fokus on SAP:lla siirtymässä palvelimen kontrolloimista sovelluksista laitteella kontrolloitaviin sovelluksiin, mikä asettaa lisähaasteita tietoturvan kannalta, koska palvelimelta sovelluksen kontrollointivastuun pois siirtäminen tarkoittaa, että kaikki suurempi vastuu esimerkiksi sovelluksen tarvitseman tiedon hakemiseen joudutaan antamaan mobiililaitteen sovellukselle. Sovelluslogiikan siirtäminen mobiililaitteelle tai mille tahansa muulle päätelaitteelle, ei kuitenkaan tarkoita, että päätelaitteelta tulevat kyselyt ovat aina luotettavia. Kyselyitä on mahdollista muokata tai tehdä kokonaan uusia, mistä syystä kyselyiden validointi täytyy aina tehdä palvelimella ennen kyselyn prosessointia tai esimerkiksi riskinä on luottamuksellisen tiedon tuhoutuminen tai joutuminen väärin käsiin.

SAP-tuotteissa on monia eri vaihtoehtoja, kuten perusoikeusautentikointi (Basic access authentication), selainperusteinen SAML 2.0, X.509 asiakassertifikaatti ja SAP kirjautumisticketti (Logon Ticket), käyttäjän tunnistamiseen. Osa näistä mahdollistaa kertakirjautumisen (SSO) eli pääsyn useisiin järjestelmiin yhdellä kirjautumisella. Kertakirjautuminen vähentää käyttäjältä tarvetta useiden salasanojen muistamiseen ja se säästää myös aikaa, kun eri järjestelmiin ei tarvitse kirjautua erikseen. Mobiilisovellusten osalta SAP suosittelee käyttämään joko x.509 asiakassertifikaattia (SUP:ia käyttäessä) tai SAML 2.0:aa (muissa tapauksissa), mutta muutkin vaihtoehdot ovat mahdollisia. (Fischer & Podgaetsky 2012)

Autorisoinneilla määritellään käyttäjän oikeudet esimerkiksi taustajärjestelmissä. Autentikoinnin ja autorisoinnin avulla voidaan esimerkiksi määritellä, mitä etäfunktiokutsuja käyttäjä voi kutsua tai mistä SAP:n tauluista hänellä on oikeus kysyä tietoja. Autorisoinnit voidaan määritellä SAP-järjestelmän lisäksi myös Netweaver Gateway -välityspalvelimella tai suoraan sovelluksissa. Mobiilisovelluksissa laitteen lähettämiin tietoihin ei kuitenkaan voida luottaa, joten käyttöoikeuksien tarkastus on syytä toteuttaa ensi sijassa palvelimella. SUP:ia tai SAP Afariaa käyttäessä käyttöoikeuksien kontrollointi on mahdollista myös suoraan laitteella esimerkiksi sovellusten käytön rajaamisen kannalta. SUP:kin käyttäessä käyttöoikeuksia tosin kontrolloidaan keskuspalvelimelta. (Fischer & Podgaetsky 2012)

4.2.3 Tiedonhallinta mobiililaitteella

Mobiililaitteella tapahtuva tiedonhallinta mahdollistaa tietoliikenneyhteyksien virheherkkyyden tai heikon laadun aiheuttamia vaikutuksia sovelluksen käytölle. Tiedonhallintaan kuuluvat niin sovelluksen käyttämien tietojen tallentaminen laitteelle välimuistiin tai pysyvään muistiin sekä tietojen synkronoiminen laitteen ja palvelimien välillä. Mobiililaitteella tietoja voidaan sovellustyypistä riippuen tallentaa joko suoraan tiedostoina tiedostojärjestelmään, tietokantaolioina tietokantaan tai esimerkiksi avain-arvo -pareina selaimen rajapintoja hyödyntäen. Myös laitteella säilytettävien tietojen turvallinen säilyttäminen on tärkeä osa tiedonhallintaa.

SAP SUP tarjoaa useita eri mobiililaitteen tiedonhallinnan ominaisuuksia, kuten tietojen ennakkoon lataamisen laitteelle, tietojen salauksen laitteella sekä muuttuneiden tietojen synkronoinnin laitteen ja palvelimen välillä. Näiden hyödyntäminen vaatii kuitenkin SUP palvelimen hankkimista ja konfiguroimista. Tiedonhallinnan ominaisuuksia on mahdollista kehittää myös ilman SUP:ia, mutta tämä voi kuitenkin olla hyvin aikaa vievää ja kustannuksiltaan kallista varsinkin, jos halutaan kehittää monikäyttöistä ratkaisua. Yksittäisille sovelluksille tiedonhallinnan toteuttaminen voi kuitenkin olla kannattavinta itse tehtynä. Tällöin voidaan esimerkiksi käyttää sovelluskehyskiä, jotka tarjoavat osittaisia ratkaisuja, joiden päälle haluttu tiedonhallintatoiminnallisuus, kuten synkronointi, voidaan toteuttaa.

HTML5:een liittyy mobiililaitteelle tiedon tallentamisen mahdollistavia toiminnallisuuksia, jotka ovat Web Storage (W3C 2011) ja Web SQL Database (W3C 2010). Näiden avulla voidaan toteuttaa periaatteessa hyvinkin monimutkaisia toteutuksia tiedon hallintaan mobiililaitteella. Web Storage määrittelee rajapinnan avain-arvo -parien tallentamiseen www-sivustokohtaisesti ja useimmat (Deveria 2012) mobiiliselaimet tukevat sitä. Web SQL Database tarjoaa Web Storagea monimutkaisemman rajapinnan tiedon tallentamiseen www-sivustojen kehittäjille, jossa tietoja tallennetaan Sqlite-tietokantaan. Myös Web SQL Database on kattavasti tuettu (Deveria 2012) eri mobiiliselaimissa.

Tiedon tallentaminen mobiililaitteelle mahdollistaa sovelluksen käyttämisen myös tietoliikenneyhteyksien katketessa. Tällöin esimerkiksi sovelluksessa voidaan ylläpitää muuttuneita tietoja ja välittää eli synkronoida ne palvelimelle, kun tietoliikenneyhteydet alkavat jälleen toimimaan. Synkronoinnin toteuttaminen vaatii erillistä sovelluskehitystä, ellei sovelluksen toteuttamisessa käytettävä sovelluskehys tarjoa valmista toteutusta synkronointiin.

5 ISO-standardit ohjelmiston laatuun liittyen

Kansainvälinen standardoimisorganisaatio International Organization for Standardization (ISO) määrittelee useita standardeja ohjelmistojen laadun määrittämiseen. Standardeja on saatavilla esimerkiksi ohjelmistojen käytettävyyden suunnitteluun (ISO 9241-210, 2010) ja näyttöpäätteiden ergonomian arviointiin (ISO 9241-11, 1998), jotka molemmat kuuluvat laajempaan ISO 9241 –standardiperheeseen, joka käsittelee tietokoneen ja ihmisen välisen interaktion ergonomiaa ja käytettävyyttä. Ohjelmistojen laadun määrittelemiseen ja arviointiin ISO:lla on myös useita standardeja. ISO/IEC 9126 -standardin 1. osa (ISO/IEC 9126-1, 2001) on tarkoitettu ohjelmiston laadun mittaamisen tueksi. Standardi sisältää myös osat 2:sta 4:ään, joissa kuvataan tarkemmin standardin 1. osassa esitettyjä laadun arviointiin liittyviä kriteereitä. ISO/IEC 14598 -standardin 1. osassa (ISO/IEC 14598-1, 1999) taas kuvataan ohjelmiston laadun arvioinnin prosessia menetelmänäkökulmasta. Sekä ISO/IEC 9126 että ISO/IEC 14598 standardit ovat korvautuneet osittain ISO/IEC 25000 (2005) SquaRE -sarjan standardeilla, joskin ISO:n mukaan vanhojakin standardeja voidaan edelleen käyttää.

ISO/IEC 9126-1 määrittelee ohjelmiston laadulle mallin. Malli koostuu kuudesta ominaisuudesta, jotka ovat: Toiminnallisuus (Functionality), Luotettavuus (Reliability), Käytettävyys (Usability), Tehokkuus (Efficiency), Ylläpidettävyys (Maintainability) ja Siirrettävyys (Portability). Ominaisuudet jakautuvat edelleen alaominaisuuksiin, joiden mukaan ohjelmiston laatua voidaan mitata. Laatumalli on esitettyinä taulukossa 1. ISO/IEC 9126-1 ei kuitenkaan tarjoa välineitä tai mittareita, joilla laatua voidaan mitata, vaan se vain määrittelee laadun eri kriteerit. ISO/IEC 9126-1:n laatumallia voidaan käyttää niin toiminnallisten kuin ei-toiminnallisten ohjelmiston ominaisuuksien laadun arviointiin sekä asiakkaan että käyttäjän näkökulmasta. Laatumallia voidaankin hyödyntää ohjelmistokehitysprojektin eri vaiheissa esimerkiksi vaatimusmäärittelyjen kokonaisvaltaisuuden tai valmiin sovelluksen hyväksyttävyyden arvioinnissa.

ISO/IEC 9126 -laatumallin tavoitteena on mahdollistaa ohjelmiston haluttu toimivuus suunnitellussa kontekstissa, joka kattaa esimerkiksi käyttöympäristön fysikaaliset tekijät ja työnteon prosessin toiminnot. Laatumallissa korostetaan, että laadun arvioinnissa on otettava konteksti huomioon kaikissa vaiheissa, kuten laatuvaatimuksia määrittäessä, laadun arvioinnin välineitä valitessa sekä laatuarviointeja tehdessä. Laatumalli määrittelee ohjelmiston laadun koostuvan sisäisistä (internal) ja ulkoisista (external) laatutekijöistä, jotka vaikuttavat ohjelmiston laadukkuuden käytössä (quality in use).

Sisäinen laatu (internal quality) kuvaa ominaisuuksien kokonaisvaltaisuutta ohjelmistotuotteen sisäisten ja muuttumattomien laatuvaatimusten näkökulmasta. Sisäisen laadun arviointikriteerit tai -mittarit eivät riipu ohjelman ajosta tai ajoympäristöstä. Ulkoinen laatu (external quality) kuvaa ominaisuuksien kokonaisvaltaisuutta ohjelmistotuotteen ulkoisten laatuvaatimusten näkökulmasta. Ulkoisen laadun arviointikriteerit tai -mittarit huomioivat ohjelman ajonaikaisen tilan. Laadukkuus käytössä (quality in use) on ohjelmiston käyttäjän näkökulma ohjelmiston laatuun, kun ohjelmistoa käytetään tietyssä ympäristössä ja kontekstissa. Tämän laatua arvioidaan sen perusteella, miten käyttäjät saavuttavat tavoitteensa tietyssä kontekstissa eli arviointi ei kohdistu suoraan ohjelmistoon.

Laatumallin tavoitteena on, että sisäisten ja ulkoisten tekijöiden tarkoituksen ja kontekstin mukaisella arvioinnilla mahdollistetaan hyvä käyttölaatu ohjelmistolle. Käyttölaatu määritellään standardissa sisäisten ja ulkoisten laatutekijöiden lisäksi mahdolliseksi arvioida esimerkiksi määrällisin mittarein. ISO/IEC 9126 ei kuitenkaan ohjeista arviointimenetelmien valinnassa tarkemmin kuin esittämällä, että arviointimenetelmien on huomioitava konteksti sekä tarkoituksenmukaisuus.

Vaikka ISO/IEC 9126 ei anna ohjeita laadun arviointimenetelmien valinnassa, se antaa kuitenkin yleisiä ohjeita, miten valituista menetelmistä tulee kertoa mallin pohjalta tehdyssä laaturaportissa. Standardi ohjeistaa, että määrällisin menetelmin ohjelmistotuotteiden arviointia tai vertailua tehdessä, raportin pitää kertoa ovatko menetelmät objektiivisia, empiirisiä ja toistettavia. Standardissa todetaan myös, että laatumallin perusteella toteutettu eri tuotteiden välinen vertailu on luotettava vain, jos arvioinnissa on käytetty tarkkoja mittareita. Tarkat mittarit ovat aina määrällisiä, koska ainoastaan niillä on mahdollista päästä tarkkoihin, yksikäsitteisiin ja objektiivisiin tuloksiin. Tarkkojen mittareiden valinnassa on haasteena löytää mittarit, jotka kuvaavat hyvin ja kattavasti oikeaa laatukriteeriä. Mittarit ja yleensäkin arviointimenetelmät riippuvat myös ohjelmistosta ja laadun arvioinnin tarpeista, joten yleispäteviä ohjeistuksia arviointimenetelmien valintaan on vaikea määritellä.

Useat ISO/IEC 9126 standardin laatutekijät ovat kuitenkin vaikeasti määrällisin menetelmin mitattavia, jos halutaan saada oikeasti kuvaavaa aineistoa. Esimerkiksi ohjelmiston opittavuutta voidaan kyllä mitata kyselylomakkeen avulla ohjelmiston käyttäjiltä, muun muassa pyytämällä käyttäjiä arvioimaan ohjelmiston opittavuutta 0-5 -asteikon avulla. Eri asia on, mittaako yksinkertainen 0-5 -arvio ohjelmiston opittavuutta kattavasti ja vaikuttaako opittavuudesta annettuihin arvioihin muut tekijät, kuten sovelluksen käyttöliittymän ulkoasun miellyttävyys. Toinen vaihtoehto voisi olla pyytää käyttäjiä arvioimaan ohjelman opittavuutta sanallisesti vapaasti muotoillen, jolloin laadullisin tutkimusmenetelmin olisi mahdollista ymmärtää syvällisemmin, mitkä tekijät vaikuttavat ohjelman opittavuuteen ja miten opittavuutta voitaisiin parantaa. Opittavuus on vain yksi ISO/IEC 9126:n laatukriteereistä, joiden arviointiin voisi monessa tutkimustilanteessa sopia paremmin laadullinen kuin määrällinen arviointimenetelmä.

Ohjelmistotuotteen arviointiprosessi arvioijan näkökulmasta kuvataan ISO/IEC 14598-5 standardissa. Standardi määrittelee arvioinnin tärkeimmät ominaisuudet, joita ovat toistettavuus (repeatability), uudelleen toteutettavuus (reproducibility), puolueettomuus (impartiality) sekä objektivisuus (objectivity). Standardi kuvaa myös arviointiprosessin eri vaiheet, joita ovat:

1. Arviointitavoitteiden selvittäminen.
2. Arvioinnin määrittely tavoitteiden ja tuotteen pohjalta.
3. Arviointisuunnitelman sekä kriteeristön luominen arviointimäärittelyyn perustuen.
4. Arvioinnin toteuttaminen arviointisuunnitelman mukaisesti.
5. Johtopäätösten tekeminen toteutetun arvioinnin perusteella.

ISO 14598-5:ssä ei kuvata yksityiskohtaisia arviointikriteerejä, koska yleispäteviä kriteerejä on vaikeaa ellei mahdotonta määritellä. Tämä johtuu standardin laajasta sovellettavuuskentästä sekä eri ohjelmistotuotteiden keskinäisistä eroavaisuuksista. Yksityiskohtaisten arviointikriteerien valinnan on kuitenkin perustuttava arviointitavoitteisiin sekä arvioitavaan tuotteeseen, jotta arviointi on standardin mukainen.

Taulukko 1. ISO 9126-1 laatumallissa määritellyt ominaisuudet sekä näiden kuvaukset ja alaominaisuudet.

Ominaisuus	Kuvaus	Alaominaisuudet
TOIMINNALLISUUS	Ohjelmiston kyky tarjota tarkoituksenmukaiset toiminnot ja täyttää halutut tarpeet, kun ohjelmistotuotetta käytetään tietyissä olosuhteissa.	Soveltuvuus
		Tarkkuus
		Yhteensopivuus
		Turvallisuus
		Toiminnallisuusohjeistuksien mukaisuus
LUOTETTAVUUS	Ohjelmiston kyky pysyä toiminnassa määritellyllä tavalla, kun ohjelmistoa käytetään määritellyllä tavalla.	Kypsyys
		Virheiden sieto
		Elpyvyys
		Luotettavuusohjeistuksien mukaisuus
KÄYTETTÄVYYS	Ohjelmiston ymmärrettävyys, opittavuus sekä sen käytön helppous ohjelmiston käyttäjän näkökulmasta, kun ohjelmistotuotetta käytetään tietyllä ennalta määritellyllä tavalla.	Ymmärrettävyys
		Opittavuus
		Operoitavuus
		Houkuttelevuus
		Käytettävyysohjeistuksien mukaisuus
TEHOKKUUS	Ohjelmiston kyky taata riittävä suorituskky suhteutettuna käytettyihin resursseihin, kun ohjelmistoa käytetään määritellyllä tavalla.	Aikakäytös
		Resurssien hyödyntämiskyky
		Tehokkuusohjeistuksen mukaisuus
YLLÄPIDETTÄVYYS	Ohjelmiston valmius mukautua esimerkiksi toimintaympäristössä tapahtuneisiin muutoksiin.	Analysoitavuus
		Muutettavuus
		Vakaus
		Testattavuus
		Ylläpidettävyysohjeistuksien mukaisuus
SIIRRETTÄVYYS	Ohjelmiston kyky toimia ympäristöstä ja ympäristön tekijöistä riippumatta.	Sopeutuvuus
		Asennettavuus
		Rinnakkaiselo
		Korvattavuus
		Siirrettävyysohjeistuksien mukaisuus

5.1 ISO/IEC 9126 Toiminnallisuus

ISO/IEC 9126 määrittelee toiminnallisuuden ohjelmistotuotteen kykynä tarjota tarkoituksenmukaiset toiminnot ja täyttää halutut tarpeet, kun ohjelmistotuotetta käytetään tietyissä olosuhteissa. Tarpeet kattavat tässä niin esitetyt kuin esittämättömätkin mutta olennaiset tarpeet. Toiminnallisuudessa näkökulmana on mitä ohjelmisto tekee tarpeiden täyttämiseksi, kun taas muilla ISO/IEC 9126 laatumallin ominaisuuksilla keskiössä on, milloin ja miten ohjelmisto täyttää tarpeet.

Toiminnallisuus jakautuu seuraaviin alaominaisuuksiin: soveltuvuus (Suitability), tarkkuus (Accuracy), yhteensopivuus (Interoperability), turvallisuus (Security) ja toiminnallisuusohjeistuksien mukaisuus (Functionality compliance). Soveltuvuus tarkoittaa ohjelmistotuotteen kykyä tarjota tarvittavat toiminnot määriteltyjen tehtävien suorittamiseksi ja käyttäjän tarpeiden täyttämiseksi. Tarkkuudella kuvataan ohjelmiston tuottamien tulosten oikeellisuutta ja sopivaa tarkkuutta ennalta määriteltyjen rajojen puitteissa. Yhteensopivuus taas kuvaa ohjelmiston kykyä toimia yhdessä yhden tai useamman muun järjestelmän kanssa. Turvallisuus määrittelee ohjelmiston kyvyn suojella tietoa riittävällä tarkkuudella esimerkiksi estämällä luottamattomien henkilöiden ja järjestelmien pääsyn käsiksi siihen. Turvallisuus alaominaisuuteen kuuluu myös, että luotetut henkilöt ja järjestelmät pääsevät käsiksi ohjelmiston tietoon tarpeenmukaisella tasolla. Toiminnallisten ohjeiden mukaisuus kuvaa ohjelmiston kykyä noudattaa toiminnallisuuteen liittyviä standardeja, mukautuneita tapoja ja säädöksiä.

5.2 ISO/IEC 9126 Luotettavuus

ISO/IEC 9126:n luotettavuus kuvaa ohjelmiston kykyä pysyä toiminnassa määritellyllä tavalla, kun ohjelmistoa käytetään määritellyllä tavalla. Luotettavuutta voidaan esimerkiksi arvioida sen perusteella, miten ohjelmisto selviää ennakoimattomista virhetilanteista. Ohjelmisto ei kulu, joten luotettavuusongelmat johtuvat esimerkiksi virheellisistä määrittelyistä, määrittelyjen toteutuksista tai ympäristössä tapahtuvista muutoksista.

Luotettavuus jakautuu seuraaviin alaominaisuuksiin: kypsyys (Maturity), virheiden sieto (Fault tolerance), elpyvyys (Recoverability) ja luotettavuusohjeistuksien mukaisuus (Reliability compliance). Kypsyys kuvaa ohjelmiston kykyä välttää ohjelman kaatumista ohjelmistovirheistä johtuen. Virheiden sieto määrittelee ohjelmiston kykyä ylläpitää määritelty suorituskyvyn taso ohjelmistovirheistä riippumatta. Elpyvyydellä taas kuvataan ohjelmistotuotteen kykyä palautua määritellylle suorituskyvyn tasolle virhetilanteen jälkeen sekä palauttaa virhetilanteessa vaikutuksen alaisena olleet tiedot. Luotettavuusohjeistuksien mukaisuus erittelee ohjelmiston kykyä noudattaa luotettavuuteen liittyviä standardeja, tapoja ja säädöksiä.

5.3 ISO/IEC 9126 Käytettävyys

ISO/IEC 9126:ssa käytettävyys on määritelty ohjelmistotuotteen ymmärrettävyytenä, opittavuutena sekä sen käytön helppoutena ohjelmiston käyttäjän näkökulmasta, kun ohjelmistotuotetta käytetään tietyllä ennalta määritellyllä tavalla. Ohjelmistotuotteen käyttäjiin lasketaan niin sovelluksen suorat käyttäjät kuin sovellusta käyttämättömät mutta sen vaikutuspiirissä tai siitä riippuvaisia olevat henkilöt. Jotkin laatumallin muut ominaisuudet, kuten toiminnallisuus ja luotettavuus, vaikuttavat myös käytettävyyteen.

Käytettävyys jakautuu seuraaviin alaominaisuuksiin: ymmärrettävyys (Understandability), opittavuus (Learnability), operoitavuus (Operability), houkuttelevuus (Attractiveness) sekä käytettävyysohjeistuksien mukaisuus (Usability compliance). Ymmärrettävyys kuvaa, miten helppo käyttäjän on ymmärtää, mihin ohjelmistoa on tarkoitus käyttää sekä millä tavoin ohjelmistoa on käytettävä, jotta määriteltyn tavoitteeseen päästään.

Ymmärrettävyys riippuu ohjelmistotuotteen dokumentaation laadusta sekä käyttäjän ensivaikutelmista ohjelmistoon liittyen. Opittavuus kuvaa, miten helppoa käyttäjän on oppia käyttämään sovellusta. Operoitavuus taas määrittää ohjelmistotuotteen kyvyn mahdollistaa käyttäjälle sovelluksen käyttö ja hallinta. Operoitavuuteen liittyy esimerkiksi mahdollisuus keskeyttää ohjelmiston muun käytön estävä pitkä ja sillä hetkellä tarpeeton ajo kesken ajon suorituksen, jolloin käyttäjän ei tarvitse odottaa ajon valmistumista päästäkseen jatkamaan ohjelmiston käyttämistä. Houkuttelevuus taas kuvaa ohjelmistotuotteen miellyttävyyttä ja puoleensavetävyyttä käyttäjän näkökulmasta. Houkuttelevuuteen vaikuttavat esimerkiksi värit, fontit ja ohjelmiston visuaalinen tyyli. Käytettävyysohjeistuksien mukaisuus kuvaa, miten ohjelmistotuote noudattaa käytettävyteen liittyviä standardeja, ohjeistuksia sekä vakiintuneita tapoja.

5.4 ISO/IEC 9126 Tehokkuus

ISO/IEC 9126:ssa tehokkuus määritellään ohjelmiston kykynä taata riittävä suorituskky suhteutettuna käytettyihin resursseihin, kun ohjelmistoa käytetään määritellyllä tavalla. Resursseihin kuuluvat esimerkiksi muut ohjelmistot, joita ohjelmistotuote hyödyntää sekä ohjelmistoalustan ohjelmat ja laitteet. Ohjelmiston tehokkuutta voidaan arvioida esimerkiksi mittaamalla, kuinka kauan siltä kestää suorittaa jokin algoritmi.

Tehokkuus jakautuu seuraaviin alaominaisuuksiin: aikakäytös (Time behaviour), resurssien hyödyntämiskyky (Resource utilisation) ja tehokkuusohjeistuksen mukaisuus (Efficiency compliance). Aikakäytös kuvaa ohjelmiston kykyä tarjota sopivat vaste- ja prosessointiajat, sen suorittaessa toimintojaan määritellyissä olosuhteissa. Resurssien hyödyntämiskyky kuvaa ohjelmistoa sen perusteella, hyödyntääkö se tarkoituksenmukaisen määrän tarkoituksenmukaisia ihmisiin liittymättömiä resursseja suorittaessaan toimintojaan määritellyissä olosuhteissa. Tehokkuusohjeistuksien mukaisuus taas kuvaa ohjelmistoa siltä osin, miten se noudattaa tehokkuuteen liittyviä standardeja, ohjeistuksia ja hyväksi havaittuja tapoja.

5.5 ISO/IEC 9126 Ylläpidettävyys

Ylläpidettävyys kuvaa ISO/IEC 9126:n mukaan ohjelmiston muutosvalmiutta. Muutoksiin kuuluvat esimerkiksi virhekorjaukset, ohjelmistoon kehitettävät uudet toiminnot sekä ohjelmiston toimintaympäristössä tapahtuneista muutoksista johtuvat olemassa olevien ohjelmiston toimintojen muutokset. Ylläpidettävyys kuvaa siis esimerkiksi, millä määrällä resursseja on mahdollista säilyttää ohjelmisto käyttökelpoisena sovelluksen muuttuvassa toimintaympäristössä.

Ylläpidettävyys koostuu seuraavista alaominaisuuksista: analysoitavuus (Analysability), muutettavuus (Changeability), vakaus (Stability), testattavuus (Testability) ja ylläpidettävyysohjeistuksien mukaisuus (Maintainability compliance). Analysoitavuus kuvaa ohjelmiston ymmärrettävyyttä esimerkiksi virhelähteiden tai muutoksia tarvitsevien ohjelman osien löydettävyyden näkökulmasta. Analysoitavuus ei liity sovelluksen ymmärrettävyyteen käyttöön liittyvän käytettävyyden näkökulmasta, vaan se liittyy ohjelmiston sisäisen rakenteen ymmärrettävyyteen. Muutettavuus liittyy ohjelmiston

muokattavuusvalmiuteen eli se kuvaa, miten hyvin ohjelmisto mahdollistaa muutosten tekemisen siihen. Muutettavuuteen voi liittyä esimerkiksi ohjelmoijan tai käyttäjän tekemät muutokset. Vakaus on määritelty ohjelmiston kyvyksi välttää odottamattomia vaikutuksia ohjelmistoon tehdyistä muutoksista johtuen. Vakaus kuvaa siis samantyyppistä ohjelmiston ominaisuutta kuin toiminnallisuus-ominaisuuden alaominaisuus virheiden sietokyky, joskin virheiden sietokyky liittyy enemmän ohjelmiston suoritukseen, kun taas vakaus liittyy enemmän ohjelmiston sisäiseen rakenteeseen. Testattavuus kuvaa ohjelmiston oikeellisuustarkistettavuutta muutosten seurauksena. Testattavuuteen liittyy esimerkiksi ohjelmiston yksikkötestattavuus. Ylläpidettävyysohjeistuksien mukaisuus kuvaa, miten se noudattaa ylläpidettävyyteen liittyviä standardeja, ohjeistuksia sekä tapoja.

5.6 ISO/IEC 9126 Siirrettävyys

ISO/IEC 9126 määrittelee siirrettävyyden ohjelmiston kyknä toimia ympäristöstä ja ympäristön tekijöistä riippumatta. Ympäristöön voi kuulu niin organisationaalisia tekijöitä kuin laitteisiin tai muihin ohjelmiin liittyviä tekijöitä. Siirrettävyydeltään laadukas ohjelmisto voi esimerkiksi olla sellainen, joka käyttää mahdollisimman vähän laitekohtaisia rajapintoja.

Siirrettävyys koostuu seuraavista alaominaisuuksista: sopeutuvuus (Adaptability), asennettavuus (Installability), rinnakkaiselo (Co-existence), korvattavuus (Replaceability) ja siirrettävyysohjeistuksien mukaisuus (Portability compliance). Sopeutuvuus kuvaa ohjelmiston kykyä mukautua erilaisiin määriteltyihin ympäristöihin ilman tarvetta määriteltyistä poikkeavien toimintojen suorittamiselle. Sopeutuvuuteen kuuluu esimerkiksi ohjelmiston käyttöliittymän sisällön skaalautuvuus erikokoisille näytöille. Asennettavuus kuvaa ohjelmiston kykyä asentua määriteltyyn ympäristöön. Rinnakkaiselo kuvaa ohjelmiston kykyä toimia yhdessä ohjelmistosta riippumattomien muiden ohjelmien kanssa yhteisessä ympäristössä, jossa ohjelmat hyödyntävät samoja resursseja. Rinnakkaiselo eroaakin toiminnallisuuden alaominaisuudesta yhteensopivuudesta, jossa eri ohjelmistot ovat yhteydessä toisiinsa. Korvattavuudella määritellään ohjelmiston kyknä korvata samaan tai vastaavaan tehtävään kehitetty ohjelmisto tietyssä toimintaympäristössä. Korvattavuuteen liittyy esimerkiksi ohjelmiston päivitettävyys siltä osin, että ohjelmiston on käyttäjän kannalta tärkeää toimia käyttäjän tietokoneella myös ohjelmiston päivittämisen jälkeen. Siirrettävyysohjeistuksien mukaisuus määrittelee, miten ohjelmisto noudattaa standardeja, muita ohjeistuksia ja hyväksi havaittuja tapoja.

6 Analyysimenetelmä

Tutkimus tehdään asiantuntija-arviointina SAP:n suosittelemista mobiilisovelluskehyksistä ISO 9126 -standardin määrittelemän laatumallin pohjalta. Tutkimus toteutetaan ISO 14598-standardin 5-osassa¹¹ (ISO/IEC 1998) määritellyn ohjelmistotuotteen arviointiprosessin mukaisesti. Asiantuntija-arviointi on tutkimusmenetelmä, jossa asiantuntija tai asiantuntijaryhmä suorittaa tuotteen tai tuotteiden arvioinnin. Asiantuntija-arviointia käytetään esimerkiksi ohjelmiston käytettävyyden arviointiin. Sillä voidaan arvioida myös muita tekijöitä, kuten dokumentaation laatua, ja sitä voidaanakin käyttää kaikissa tuotteen suunnittelu- ja kehitysvaiheissa (Korvenranta 2005). Asiantuntija-arviointi toteutetaan vertaamalla tuotetta tai tuotteita määriteltäviin ohjeistuksiin tai periaatteisiin, jotka pyritään tekemään mahdollisimman objektiivisiksi. Kuitenkin asiantuntija-arviointi on aina jollain tasolla subjektiivista eli arvioinnin tulokset riippuvat arvioinnin tekijästä. Ennen arvioinnin toteutusta on hyvä perehtyä arvioitavien tuotteiden käytön kontekstiin, jotta asiantuntija-arvioinnissa voidaan keskittyä tärkeimpiin tekijöihin. (Nielsen 2006)

Tutkimuksessa aineistoa analysoidaan tarkastellen sovelluskehyskiä mobiilin tietojenkäsittelyn kontekstiin pohjautuen käyttäen ISO 9126 mukaista laadun arviointia. Esimerkiksi sovellustoiminnallisuuksien toteutettavuutta sovelluskehyskiillä arvioidaan ISO 9126 pohjalta. Arviointi toteutetaan antamalla ISO 9126 määrittelemien 6 ominaisuuden ja näiden alaominaisuuksien pohjalta sovelluskehyskiille plussia ja miinuksia. Arviointi perustuu sovelluskehysten kykyyn mahdollistaa mobiilin SAP-paikkatietosovelluksen kontekstiin liittyvien laadukkaiden toiminnallisuuksien toteutettavuus.

Arviointi toteutetaan analysoiden sovelluskehyskiä laadullisella menetelmällä esimerkiksi dokumentaatioiden ja olemassa olevien sovellusten pohjalta. Plussia ja miinuksia annetaan ominaisuuskohtaisesti verraten muihin tutkimuksen sovelluskehyskiin, mikäli tämä on mahdollista. Jossain tapauksissa vertailu sovelluskehysten välillä voi olla vaikeaa, jolloin vertailussa pyritään huomioimaan mahdollisia yleisiä laatuominaisuuksiin liittyviä kriteereitä. Muihin sovelluskehyskiin verrattaessa ei tavoitella objektiivisin mittarein tehtävää vertailua, vaan tavoitteena on tarkastella laatuominaisuutta laajempänä kokonaisuutena, jota pyritään kuvaamaan kokoamalla plussia ja miinuksia sovelluskehysten tutkimuksen perusteella laatuun vaikuttavista tekijöistä.

6.1 Analyysikriteeristöä yleisesti

Tutkimuksessa arvioidaan ohjelmistotuotteiden laatua ISO 9126:ssa määritellyn kuudesta ominaisuudesta koostuvan laatumallin mukaisesti. Analysointi toteutetaan määrittelemällä kuuden laatuominaisuuden valituille alaominaisuuksille kriteerit, joiden perusteella ohjelmistotuotteiden laatua arvioidaan toisiin ohjelmistotuotteisiin verraten. Kriteerit on valittu mobiilin SAP-paikkatietosovelluksen kontekstin pohjalta, joten tutkimuksen pohjalta tehtyä laatuarviointia ei voida suoraan soveltaa muihin konteksteihin. Kriteerien valinnassa on pyritty löytämään tutkimuksen tavoitteiden kannalta parhaiten ohjelmistokehysten laatua kuvaavat alaominaisuudet. Arvioinnin ulkopuolelle on jätetty laadun alaominaisuuksia, joita ei ole katsottu tutkimuksen tavoitteiden kannalta relevanteiksi. Näistä on perusteltu, miksi niitä ei ole valittu tutkimukseen.

¹¹ ISO 14598-standardin 5. osa kuvaa ohjelmistotuotteen arviointiprosessin arvioijan näkökulmasta.

6.2 Toiminnallisuuden kriteerit

Toiminnallista laatua arvioidaan sen alaominaisuuksien soveltuvuuden, tarkkuuden, yhteensopivuuden ja turvallisuuden kautta. Toiminnallisuusohjeistuksien mukaisuutta ei arvioida, koska toiminnallisuusohjeiden on katsottu olevan usein enemmän tiettyyn alustaan tai käyttötapaukseen, kuten teollisuuden prosessiin, kuin sovelluskehyksiin liittyviä.

Sovelluskehysten toiminnallista laatua soveltuvuuden näkökulmasta arvioidaan mobiilin SAP-paikkatietosovelluksen olennaisten toiminnallisuuksien osalta. Soveltuvuutta on lähestytty toteutettavuuden näkökulmasta, koska tutkimuksella on haluttu selvittää, miten eri sovelluskehykset soveltuvat erityisesti mobiiliin SAP-paikkatietokontekstin sovellusten kehittämiseen. Arvioinnissa keskitytään paikkatietosovellustoiminnallisuuksien (kartta, paikannus, reititys ja paikkatietoanalyysit) toteutettavuuteen. SAP-sovellustoiminnallisuuksista käsitellään ainoastaan mobiililaitteella tapahtuvaan tiedonhallintaan liittyviä ominaisuuksia, koska muut olennaiset SAP-sovellustoiminnallisuudet käsitellään muiden alaominaisuuksien yhteydessä.

Sovelluskehysten tarkkuutta laadun kannalta tarkastellaan sen perusteella mihin käyttöön sovelluskehykset on suunniteltu. Ensimmäisenä tarkkuuden kriteerinä on, onko sovelluskehykset suunniteltu erityisesti mobiilisovellusten tarpeisiin vai ovatko ne kenties suunniteltu sekä mobiili että työpöytäsovellusten toteuttamiseen. Mobiiliuden näkökulmasta suunnitellut sovellukset esimerkiksi huomioivat paremmin mobiiliuteen kiinteästi liittyvän käyttäjän muuttuvan sijainnin tarjoamalla ratkaisuja tämän huomioimiseen. Tämän lisäksi tutkitaan ovatko sovelluskehykset suunniteltu SAP-sovellusten toteuttamiseen, vai ovatko ne kenties geneerisempiä ja vähemmän integroituja SAP-alustaan.

Laadullista yhteensopivuutta arvioidaan ensisijaisesti sen perusteella, miten ja millä tavoin sovelluskehyksellä on mahdollista olla vuorovaikutuksessa tai yhteydessä SAP-järjestelmiin ja paikkatietojärjestelmiin. SAP-järjestelmien vuorovaikutuksen tarkastelun osalta tukeudutaan Yhdistettävyyden SAP-dataan luvussa esitettyihin ja SAP:n suosittelemiin vuorovaikutusmenetelmiin, kuten MBO ja OData. Yhteyksiä paikkatietojärjestelmiin tarkastellaan sen perusteella, onko sovelluskehyksessä itsessään tai onko siihen saatavilla tuki esimerkiksi Open Geospatial Consortium:n (OGC) määrittelemiin Web Map Service¹² (WMS) tai Web Feature Service¹³ (WFS) palvelurajapintoihin tai Esri:n määrittelemään Esri REST API¹⁴ palvelurajapintaan.

Sovelluskehysten turvallisuutta tarkastellaan sen mukaan, miten kattavasti ne tarjoavat turvaa tiedolle sen matkalla palvelimelta laitteelle ja takaisin. Esimerkiksi tukeeko sovelluskehys salatun yhteyden käyttöä tiedon välityksessä. Tämän lisäksi tarkastellaan sovelluskehysten tarjoamia ratkaisuja tiedon suojaamiseksi laitteella. Tarkastelussa on muun muassa, säilyttääkö sovellus tietoa laitteella ja jos säilyttää, niin ovatko tiedot

¹² Web Map Service (WMS) on HTTP-pohjainen palvelurajapinta karttakuvien hakemiseen palvelimilta. Viitattu 3.10.2012. Saatavissa: <http://www.opengeospatial.org/standards/wms>.

¹³ Web Feature Service (WFS) on HTTP-pohjainen palvelurajapinta esimerkiksi paikkatietoaineistojen hakemiseen palvelimilta. Viitattu 3.10.2012. Saatavissa: <http://www.opengeospatial.org/standards/wfs>.

¹⁴ Esri REST API on Esri:n tarjoama avoin palvelurajapinta, jolla ArcGIS Server –palvelimella oleviin palveluihin pääsee käsiksi. Viitattu 3.10.2012. Saatavissa: <http://resources.esri.com/help/9.3/arcgisserver/apis/rest/>.

salattuja. Tietoturvaa analysoidaan lisäksi sen perusteella, millainen tuki sovelluskehysissä on SAP-käyttöoikeus ja autentikointi -menetelmille, kuten SSO.

Taulukko 2. Toiminnallisen laadun arviointikriteerit.

TOIMINNALLISUUS	
Alaominaisuus	Arviointikriteeri(t)
Soveltuvuus	<ul style="list-style-type: none">• Mitä toiminnallisuutta sovelluskehys tarjoaa mobiilisovelluskehityksen tueksi:<ul style="list-style-type: none">○ SAP-sovellustoiminnallisuuksien näkökulmasta?○ Paikkatiedon sovellustoiminnallisuuksien näkökulmasta?
Tarkkuus	<ul style="list-style-type: none">• Onko sovelluskehys suunniteltu mobiilisovellusten vai muiden sovellusten toteuttamiseen?• Onko sovelluskehys suunniteltu erityisesti SAP-sovelluksien näkökulmasta?
Yhteensopivuus	<ul style="list-style-type: none">• Miten sovelluskehyksellä on mahdollista olla vuorovaikutuksessa:<ul style="list-style-type: none">○ SAP:iin○ Paikkatietojärjestelmiin (esimerkiksi paikkatietoanalyysijä varten)
Turvallisuus	<ul style="list-style-type: none">• Miten sovelluskehys tukee tietoturvaa tiedon matkalla palvelimelta laitteelle ja takaisin?<ul style="list-style-type: none">○ Tiedonsiirrossa○ Laitteella○ SAP-Käyttöoikeuksien kannalta?

6.3 Luotettavuuden kriteerit

Laadun luotettavuusominaisuuksia arvioidaan seuraavien luotettavuuden alaominaisuuksien avulla: virheiden sieto ja elpyvyys. Kypsyyttä ei arvioida, koska sovelluskehystä enemmän virhetilanteilta välttymiseen katsotaan vaikuttavan ajoympäristön toteutus, kuten selaimen Javascript-moottori. Kypsyyden katsotaan myös kuuluvan paremmin varsinaisten sovellusten kuin sovelluskehysten arviointiin, koska mahdollisia virhetilanteita on vaikea määritellä ilman sovelluksia. Luotettavuusohjeistuksien mukaisuutta ei arvioida, koska luotettavuusohjeistuksien on arvioitu sopivan paremmin yksittäisten tiettyyn tarkoitukseen kehitettyjen sovellusten laadun arviointiin kuin sovelluskehityksen apuna käytettävien sovelluskehysten laadun arviointiin.

Virheiden sietoa analysoidaan tarkastelemalla sovelluskehysistä siitä näkökulmasta, onko niissä sisäänrakennettua automaattista virhetilanteiden estoa ja virheen hallintaa. Virheiden sietoa arvioidaan sen perusteella, miten kehitetty sovellus käyttäytyy virhetilanteen

jälkeen. Esimerkiksi kaatuuko sovellus virhetilanteen jälkeen tai osaako se välttää määrittelemättömään tilaan joutumista. Virheiden sieto on myös paljolti riippuvainen sovelluslustoan toteutuksesta. Näin ollen sovelluskehysillä virheiden sieto liittyy ennen kaikkea niillä kehitettyjen sovellusten epätarkoituksenmukaiseen käyttöön ja siihen huomioiko sovelluskehys tämän.

Elpyvyyttä tarkastellaan sovelluskehysten näkökulmasta sen perusteella, miten niillä kehitetyt sovellukset toimivat virhetilanteen tai määrittelemättömään tilaan joutumisen jälkeen. Esimerkiksi palautuuko sovellus virheen tai muun sovelluksen normaalista toiminnasta poikkeavan tapahtuman jälkeen alkuperäiseen tilaan, vai jääkö se määrittelemättömään tilaan tai kaatuu. Määrittelemättömään tilaan jäävässä sovelluksessa saattaa esimerkiksi vain osa käyttöliittymästä reagoida käyttäjän syötteisiin. Elpyvyyden kannalta on hyvä, mikäli sovelluskehys palauttaa sovelluksen virhetilannetta edeltäneelle näkymälle. Elpyvyyteen vaikuttaa myös mahdolliset sovelluskehysten mekanismit, jotka suojaavat tietoja häiriötilanteissa.

Taulukko 3. Luotettavuuden arviointikriteerit.

LUOTETTAVUUS	
Alaominaisuus	Arviointikriteeri(t)
Virheiden sieto	<ul style="list-style-type: none">• Onko sovelluskehyksessä sisäänrakennettua automaattista virheen hallintaa ohjelmavirheen tapahtuessa?
Elpyvyys	<ul style="list-style-type: none">• Mihin tilaan sovellus ajautuu virheen jälkeen? Latautuuko sovellus esimerkiksi uudestaan alkutilaan vai virhettä edeltäneelle näkymälle vai jatkaako se toimintaa määrittelemättömässä tilassa?• Onko virhetilanteessa mekanismeja, jotka estäisivät tiedon häviämisen esimerkiksi varmuuskopioinnin ja tietojen palauttamisen kautta?

6.4 Käytettävyyden kriteerit

Käytettävyyttä arvioidaan seuraavien alaominaisuuksien avulla: ymmärrettävyys, opittavuus ja käytettävyysohjeistuksien mukaisuus. Houkuttelevuutta ei arvioida, koska sen katsotaan kontekstiltään liittyvän enemmän varsinaisten sovellusten käyttöön kuin sovelluskehitykseen. Houkuttelevuutta ei myöskään ole mielekästä arvioida esimerkiksi sovelluskehysten integroitujen kehitysympäristöjen (IDE) näkökulmasta, koska samoja kehitysympäristöjä voidaan käyttää useiden sovelluskehysten kanssa.

Käytettävyyden alaominaisuutta operoitavuutta ei myöskään analysoida tutkimuksessa. Sen on katsottu soveltuvan paremmin kehitettyjen sovellusten kuin sovelluskehysten arviointikriteeriksi. Operoitavuutta voitaisiin analysoida kuitenkin myös sovelluskehityksen näkökulmasta esimerkiksi arvioimalla, miten hallittava sovelluksen kehitysprosessi on sovelluskehysillä kehitettäessä. Tällainen analyysi soveltuisi kuitenkin paremmin integroitujen kehitysympäristöjen kuin sovelluskehysten analysointiin, ja tiettyä

integroitua sovelluskehitysympäristöä ei ole haluttu sitoa analyysissä kiinteästi tiettyyn sovelluskehikseen.

Sovelluskehysten ymmärrettävyyttä arvioidaan niiden mobiilisuuntaituneisuuden perusteella. Lisäksi arvioidaan, onko sovelluskehikset kehitetty ensisijaisesti SAP- tai paikkatietosovelluksien toteuttamiseen. Analyysi tehdään ensisijaisesti sovelluskehysten pääasiallisten dokumentaatioiden, sovellusesimerkkien, kehittäjän oppaiden ja muiden opastusten perusteella arvioiden, miten mobiilius, SAP ja paikkatieto on otettu huomioon sekä onko erityisesti niitä painotettu.

Opittavuutta tarkastellaan sovelluskehikseen saatavilla olevan avun näkökulmasta. Tarjoavatko sovelluskehiksen kehittäjät tai muut tahot opastuksia (tutorial) tai muita ohjeita, joilla tuotetta voi opetella käyttämään? Entä onko sovelluskehikseen saatavilla esimerkiksi virallista tukipalvelua tai vertaistukea? Opittavuutta arvioidaan ensisijaisesti suhteuttaen saatavilla oleva tuki mobiilien SAP- tai paikkatietosovellusten toteuttamiseen.

Käytettävyysohjeistuksien mukaisuutta tarkastellaan sen perusteella, miten sovelluskehikset huomioivat mobiilialustakohtaiset käytettävyysohjeistukset, kuten iOS Human Interface Guidelines¹⁵ tai Android User Interface Guidelines¹⁶. Arvioitavana on esimerkiksi, mukautuvatko sovelluskehiksellä toteutettujen sovellusten käyttöliittymät alustan käyttöliittymäohjeistuksien mukaisiksi tai huomioiko sovelluskehys alustakohtaiset tavat toteuttaa toiminnallisuudet. Käytettävyysohjeistuksien mukaisuus on tärkeää, jotta käyttäjät voivat käyttää sovellusta totutulla ja yhteisellä, alustan ohjeistuksissa määritellyllä tavalla. Esimerkiksi Android:ssa ja BlackBerry:ssä edelliselle sivulle tai edelliseen vaiheeseen siirtyminen tapahtuu fyysisestä napista, kun taas iPhone:ssa sama tapahtuu virtuaalisesta napista. Myös kontekstivalikon sijainti vaihtelee eri laitteilla. iPhone:ssa se sijaitsee laitteen oikeassa alakulmassa, kun taas Androidissa samassa kohtaa voi olla esimerkiksi takaisin-nappi. Jos sovellusta ei mukauteta jokaisen alustan käyttöliittymäohjeistuksien mukaiseksi, on riskinä, että käyttäjät painavat väärää nappia, koska ovat tottuneet, että painamassaan kohdassa on yleensä jokin toinen virtuaalinen tai fyysinen nappi (Charland&Leroux, 2011).

¹⁵ iOS Human Interface Guidelines tarjoaa käyttöliittymäohjeistuksia iOS-mobiilisovellusten tekemiseen. Viitattu 3.10.2012. Saatavissa:

<http://developer.apple.com/library/ios/#DOCUMENTATION/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html>.

¹⁶ Android User Interface Guidelines tarjoaa käyttöliittymäohjeistuksia Android-mobiilisovellusten toteuttamiseen. Viitattu 3.10.2012. Saatavissa:

http://developer.android.com/guide/practices/ui_guidelines/index.html.

Taulukko 4. Käytettävyyden arviointikriteerit.

KÄYTETTÄVYYS	
Alaominaisuus	Arviointikriteeri(t)
Ymmärrettävyys	<ul style="list-style-type: none">• Onko sovellusesimerkkejä saatavilla sovelluskehityksen käytöstä mobiilisovelluskehitykseen SAP:n ja paikkatiedon näkökulmasta?• Millaiset ovat sovelluskehityksen API-dokumentaatio ja muu dokumentaatio, kuten mahdolliset kehittäjän oppaat, mobiilisovellusten kehityksen näkökulmasta?
Opittavuus	<ul style="list-style-type: none">• Millaista tukea (opastusohjelmia, tukipalvelua, vertaistukea tms.) sovelluskehityksille on saatavilla mobiilikehityksen tueksi ja onko tukea saatavilla SAP- tai paikkatietosovellusten kehittämiseen?
Käytettävyysohjeistuksien mukaisuus	<ul style="list-style-type: none">• Huomioiko sovelluskehitys eri alustojen käyttöliittymäohjeistuksia eli esimerkiksi mukauttaako se käyttöliittymiä alustojen ohjeistuksien mukaisiksi.

6.5 Tehokkuuden kriteerit

Sovelluskehityksiä ei arvioida tehokkuuden näkökulmasta tutkimuksessa, koska tehokkuutta on vaikea analysoida staattisesti. Tehokkuuden alaominaisuudet liittyvät usein laatuun käytössä, jolloin ne ovat enemmän riippuvaisia varsinaisesta sovellustoteutuksesta kuin sovelluskehityksen ominaisuuksista. Eri sovelluskehityksillä tehdyt sovellukset voivat olla aikakäytökseltään tai resurssien hyödyntämiskyvyltään hyvin lähellä toisiaan, jolloin varsinaista tehokkuutta tärkeämmäksi muodostuvat muut tekijät, kuten käytön kannalta riittävän nopea toimintojen suoritus sopivalla resurssien kulutuksella. Tietyllä sovelluskehityksellä toteutettu sovellus saattaa myös olla tehokkuudeltaan erilainen riippuen alustasta. Esimerkiksi Javascript:nä ajettavissa sovelluksissa voi olla suorituskyvyltään suuria eroja yksinkertaisissakin toiminnoissa, vaikka sovelluskehityksen sisältämät toiminnot onkin pyritty suunnittelemaan eri alustoilla tehokkaasti toimiviksi.

Sovelluskehysten tehokkuuden analysointi käytön kannalta vaatisi käytännössä usean määritellyt mobiili-, SAP- ja paikkatietotarpeet täyttävän sovelluksen toteuttamista eri sovelluskehityksillä, jotta tehokkuutta voitaisiin arvioida kuvaavasti. Tämä olisi jo pelkästään oman tutkimuksen arvoinen kokonaisuus omine tutkimuskysymyksineen, -tavoitteineen ja -menetelmineen sekä johtopäätöksineen. Kuvaavan tehokkuustarkastelun vaatimasta työmäärästä johtuen, se on jätetty kokonaan pois tästä tutkimuksesta. Tehokkuustarkastelun pois jättämistä tukee myös se, että tutkimuksen analyysissä on ollut tarkoituksena arvioida sovelluskehityksiä staattisesti sen sijaan että analysoitaisiin niillä toteutettuja sovelluksia käytössä.

6.6 Ylläpidettävyyden kriteerit

Ylläpidettävyyteen liittyen sovelluskehysten laatua arvioidaan analysoitavuuden, muutettavuuden ja testattavuuden näkökulmasta. Ylläpidettävyysohjeistuksien mukaisuutta ei arvioida, koska ylläpidettävyysohjeistuksien, kuten testausohjeiden, on katsottu sopivan paremmin yksittäisten sovellusten kuin sovelluskehysten vertailuun.

Myöskään sovelluskehysten laatua vakauden näkökulmasta ei analysoida tutkimuksessa. Vakautta voitaisiin analysoida esimerkiksi toteuttamalla sovelluskehysiin laajennuksia ja arvioiden toteuttamisprosessissa esiin tulleita tekijöitä. Tämä edellyttäisi kuitenkin useita eri kontekstiin sopivia laajennusesimerkkejä, joiden toteuttamisen vaatiman työmäärän on katsottu olevan tutkimuksen tavoitteiden kannalta liian suuri. Myöskään ohjelmakoodin staattista analyysiä vakauden kannalta ei ole arvioitu järkeväksi, koska staattisia vakautta kattavasti ja yhdenmukaisesti eri sovelluskehyksillä mittaavia työkaluja ei ole arvioitu olevan olemassa.

Analysoitavuus-alaominaisuutta tutkitaan sovelluskehysten lähdekoodin avoimuuden ja saatavuuden sekä arkkitehtuurien dokumentoinnin näkökulmasta. Lähdekoodin vapaus esimerkiksi helpottaa sovelluskehysten sisäisen toiminnan ymmärtämistä. Dokumentointeja verrataan toisten sovelluskehysten dokumentointeihin selvittäen, miten selkeästi eri sovelluskehysten arkkitehtuuri on kuvattu. Arkkitehtuurin analysoitavuusanalyysin osalta ei kuitenkaan oteta kantaa, miten eri arkkitehtuurit varsinaisesti soveltuvat mobiilisovellusten toteuttamiseen, mikä kuuluu toiminnallisen laadun soveltuvuus-alaominaisuuteen.

Laatua muutettavuuden näkökulmasta tarkastellaan sovelluskehysten laajennuskohteiden dokumentoinnin perusteella esimerkiksi dokumentaatioissa tarjolla olevien laajennusesimerkkien näkökulmasta. Muutettavuuteen liittyy myös sovelluskehysten osien, kuten yksittäisten luokkien, ymmärrettävyys ja laajennettavuus, mutta näitä ei ole arvioitu johtuen staattisen arvioinnin vaikeudesta. Haasteita aiheuttaa esimerkiksi vertailukelpoisia tuloksia tuottavien työkalujen puute.

Testattavuutta arvioidaan tutustumalla, millaisia koodin laatua virheiden hallinnan näkökulmasta parantavia ominaisuuksia tai työkaluja sovelluskehysiin sisältyy. Sisältyykö niihin työkaluja esimerkiksi yksikkötestauksen tai muun automaattisen testauksen tueksi. Lisäksi arvioidaan, miten hyvin kehitettyjä sovelluksia on mahdollista testata niiden oikeassa tai oikeaa vastaavassa käyttöympäristössä.

Taulukko 5. Ylläpidettävyyden arviointikriteerit.

YLLÄPIDETTÄVYYS	
Alaominaisuus	Arviointikriteeri(t)
Analysoitavuus	<ul style="list-style-type: none">• Onko sovelluskehityksen lähdekoodi avointa?• Kuinka selkeästi sovelluskehityksen arkkitehtuuri on dokumentoitu?
Muutettavuus	<ul style="list-style-type: none">• Onko mahdollisia laajennuskohteita dokumentoitu ja kuvattu esimerkein?
Testattavuus	<ul style="list-style-type: none">• Sisältyykö kehikseen testauksen työkaluja ja onko testaus mahdollista oikeassa käyttöympäristössä?

6.7 Siirrettävyyden kriteerit

Sovelluskehysten laatua siirrettävyyden näkökulmasta arvioidaan sopeutuvuuden, asennettavuuden ja korvattavuuden osalta. Laatua rinnakkaiselon tai siirrettävyysohjeistuksien mukaisuuden osalta ei arvioida. Rinnakkaiselo on jätetty arvioinnin ulkopuolelle, koska laadukkaan rinnakkaiselon järjestäminen eri sovelluksille on mobiililaitteilla usein käyttöjärjestelmän sekä rinnakkaiselon huomioivien kehityskirjastojen tehtävä. Esimerkiksi älypuhelimissa käyttöjärjestelmä jakaa eri prosesseille laskenta-aikaa vapauttaen sovelluskehittäjän yksityiskohtaiselta laitteen resurssien hallinnalta. Siirrettävyysohjeistuksien mukaisuutta ei arvioida, koska näiden on katsottu sopivan paremmin sovelluskehityksillä kehitettävien sovellusten arviointiin kuin itse sovelluskehysten arviointiin.

Laatua sopeutuvuuden kannalta arvioidaan sen perusteella, tunnistavatko sovelluskehityksellä kehitetyt sovellukset sovellusalustan, eli esimerkiksi mobiililaitteen käyttöjärjestelmän sekä fyysiset näppäimet, ja sopeuttavatko ne toimintaansa alustaan sopivaksi. Sopeutuvuuden osalta ei arvioida käyttöliittymien sopeutuvuutta ohjeistuksien näkökulmasta, jota analysoidaan käytettävyysohjeistuksien mukaisuudella, vaan arvioinnissa keskitytään muiden esimerkiksi fyysisten ja teknisten toiminnallisuuksien, kuten laitteen koon sekä sen näytön tarkkuuden tai sille ominaisten nappien, huomioimiseen sovelluskehityksissä.

Asennettavuus-laatua tutkitaan sovelluskehityksillä kehitettävien sovellusten alustavaihtoehtojen ja alustaominaisuuksien näkökulmasta. Tutkimuksessa selvitetään, mille eri mobiililaitteille ja -alustoille sovelluskehityksillä on mahdollista kehittää. Vaikka sovelluskehitys mahdollistaisi sovelluskehityksen tietylle alustalle, arvioidaan laatua asennettavuuden näkökulmasta myös sen perusteella, onko eri sovelluskehityksillä eri alustoille kehitetyillä sovelluksilla alustakohtaisia rajoitteita esimerkiksi mahdollisten toiminnallisuuksien suhteen. Sovelluskehityksiä arvioidaan myös sen perusteella, onko niillä kehitetyissä sovelluksissa eri alustojen välisiä tarpeettomia eroja sovellusten toiminnallisuuksissa.

Korvattavuutta arvioidaan sovelluskehyksellä kehitettävien sovellusten päivitettävyyden näkökulmasta. Sisältääkö sovelluskehys esimerkiksi työkalut automaattiseen versiointiin sekä tukeeko se sovellusten versiopäivityksiä, mikä mahdollistaa esimerkiksi eri versioiden yhtäaikaisen kehityksen.

Taulukko 6. Siirrettävyyden arviointikriteerit.

SIIRRETTÄVYYS	
Alaominaisuus	Arviointikriteeri(t)
Sopeutuvuus	<ul style="list-style-type: none">• Osaavatko sovelluskehyksellä toteutetut sovellukset tunnistaa alustan ja mukautua sen perusteella?
Asennettavuus	<ul style="list-style-type: none">• Mille eri mobiililaitteille ja -alustoille sovelluskehyksillä voidaan kehittää sovelluksia?<ul style="list-style-type: none">○ Onko sovelluskehyksillä kehitetyillä sovelluksilla alustojen välisiä eroja? Mitkä alustojen toiminnallisuudet ovat mahdollisia?
Korvattavuus	<ul style="list-style-type: none">• Onko sovelluskehyksessä työkaluja sovellusten hallintaan päivitettävyyden näkökulmasta? Jos on, niin millaisia ne ovat?

7 Aineisto

Taulukko 7 esittelee tutkimukseen valitut sovelluskehitystyökalut. Siinä kerrotaan myös, mitä eri mobiilisovellustyyppettä on mahdollista toteuttaa valituilla työkaluilla.

Taulukko 7. Tutkimukseen valitut sovelluskehitystyökalut sekä niiden mahdollisuudet toteuttaa eri mobiilisovellustyyppettä. (vrt. Faulk 2012)

Sovelluskehitystyökalu	Web-sovellus	Hybridisovellus	Natiivisovellus
SAP UI5	X		
PhoneGap		X	
Appcelerator: Titanium Development Platform / Titanium Mobile SDK	X	X	X
Sencha Touch	X	X	
Sybase Mobile SDK		X	X

Tutkimuksen aineistona on SAP:n suosittelemia, tarjoamia tai tukemia mobiilisovelluskehitystyökaluja. Näistä on valittu tutkimukseen ajankohtaisimmat ja viime aikoina eniten huomiota esimerkiksi blogeissa (Moy 2012) sekä SAP:n viestintätiedotteissa (SAP 2012c) saaneet kehitystyökalut. Tutkimuksen ulkopuolelle on rajattu työkalut, joiden ei ole katsottu olevan tällä hetkellä ajankohtaisia, vaikka ne olisivatkin jo yleisesti käytössä sekä niille olisi luvattu tuki useaksi vuodeksi SAP:n puolelta (Esimerkiksi SAP Netweaver Mobile (Kaluscha 2011)). Tutkimuksen ulkopuolelle on myös jätetty sovelluskehitystyökalut, joita SAP:n osalta ei ole selkeästi suositeltu tai tuettu. Tämä syystä, että mobiilisovelluskehitystyökaluja on hyvin paljon ja yleisesti ottaen työkalujen osalta on vaikea sanoa, mitkä niistä saavat eniten huomiota esimerkiksi vuoden päästä, koska työkalut kehittyvät nopeasti.

Tutkimuksessa sovelluskehitystyökaluja tarkastellaan yksittäisinä ja irrallisina toisistaan, vaikka niiden rinnakkainen käyttö olisikin mahdollista tai hyödyllistä. Tähän on päädytty, jotta eri työkaluja voitaisiin paremmin vertailla keskenään. Työkalut ovat usein myös korvattavissa jollain toisella, joskin tämä vaatii käytännössä aina muutoksia, joten arvioinnit erillisinä työkaluina tehden on parempi huomioida työkaluille ominaiset piirteet. Vaihtoehtoisesti tutkimuksessa olisi voitu lähestyä arviointia enemmän mobiilisovellustyyppien näkökulmasta. Tällöin kuitenkin fokus olisi siirtynyt pois työkalujen keskinäisestä vertailusta enemmän käyttötapausten vertailun suuntaan.

7.1 SAP UI Development Toolkit for HTML5

SAP UI Development Toolkit for HTML5 (SAP UI5) on työkalu edustasovellusten (front-end application) toteuttamiseen SAP-ympäristössä. SAP UI5:llä voidaan toteuttaa

sovelluksia työpöytätietokoneella, tabletilla ja älypuhelimilla ajettavia sovelluksia. SAP UI5 on SAP:n tarjoama ratkaisu avoimien HTML5-teknologioiden hyödyntämiseen sovelluskehityksessä ja se tarjoaa vartenotettavan ratkaisuvaihtoehdon WebDynpro-tuotteille edustasovelluskehitystyökaluna SAP-ympäristössä. (Ganz 2012)

SAP UI5 eroaa aikaisemmista SAP:n edustasovellusteknologioista olemalla ensimmäinen varsinainen teknologia, jolla sovellukset tehdään ulkoa-sisään-mallilla sisältä-ulos-mallin sijaan. Sisältä-ulos-mallissa käyttöliittymä ja sen sisältö määritellään esimerkiksi XML:llä konfiguroimalla SAP:n sisällä ja se syötetään edustasovellukselle valmiiksi esitettävässä muodossa. Myös sovelluksen käyttöliittymän toiminta kierrätetään taustajärjestelmän kautta, joka luo komentojen perusteella uuden näkymän edustasovellusta varten. Sisältä-ulos-mallia hyödyntävät SAP:n tuotteista muun muassa WebDynpro for JAVA sekä Business Server Pages (BSP). Ulkoa-sisälle-mallissa taas sovellukset rakennetaan itsenäisinä sovelluksina, jotka hakevat tiedon SAP:sta tarvittaessa. Palvelin ei siis tällöin kontrolloi edustasovellusta, vaan edustasovelluksessa määritellään, miten sovellus toimii. Käännä sisältä-ulos-mallista ulkoa-sisälle-malliin on seurausta SAP:n halusta hyödyntää mahdollisimman paljon olemassa olevia web-kehittäjiä ilman, että heidän tarvitsee opetella SAP-spesifejä kehitystyökaluja, kuten WebDynpro for JAVA tai WebDynpro for ABAP. (Adams 2012)

SAP UI5 on kehitetty työpöytä-, tabletti- ja älypuhelinsovellusten toteuttamista varten. Se sisältää jQuery-kirjaston(<http://jquery.com/>), joka on yksi suosituimmista Javascript-kirjastoista web-sivujen ja –sovellusten kehittämisen tehostamiseen. Lisäksi SAP UI5 mahdollistaa muidenkin Javascript-kirjastojen hyödyntämisen. SAP UI5:n mobiilikkehitysversiota ei kuitenkaan ole vielä julkaistu¹⁷. Tähän mennessä julkaistulla SAP UI5:llä ei kaikkia toiminnallisuuksia ole vielä optimoitu eri alustoille tai toiminnallisuudet eivät ole käytettävissä esimerkiksi älypuhelimilla. Suurin osa toiminnallisuuksista kuitenkin toimii jo alustasta riippumatta. (Adams 2012)

SAP UI5 on valittu tutkimukseen, vaikka se on vielä varsin uusi ja kaikkia luvattuja toiminnallisuuksia, kuten älypuhelimille optimoitua versiota, ei vielä ole saatavilla. Se on valittu, koska sen on katsottu olevan vartenotettava tuote tulevaisuuden edustasovellusten toteuttamiseksi SAP-ympäristössä. Edustasovellukset kattavat sekä mobiili- että työpöytäsovellukset. SAP UI5:n etuna on, että samalla tekniikalla voidaan kehittää niin mobiili- kuin työpöytäsovelluksiakin, jolloin esimerkiksi saatetaan säästää kustannuksissa.

7.2 Adobe PhoneGap

Adobe PhoneGap (nykyisin Apache Cordova) on avoimen lähdekoodin sovelluskehys, jolla on mahdollista muuttaa web-sovellus mobiilihybridisovellukseksi, joka pyörii mobiililaitteelle asennetulla sovellusalustalla. Adobe PhoneGap mahdollistaa natiivien laiteominaisuuksien, kuten kameran ja paikannuksen, hyödyntämisen sovelluksesta geneeristen Javascript-rajapintojen avulla ilman, että sovelluskehittäjän tarvitsee tietää millä laitteilla sovellusta tullaan käyttämään. Adobe PhoneGap mahdollistaa hybridikehityksen HTML:llä, CSS:llä ja Javascript:illä, mutta ei kuitenkaan tarjoa näitä varten esimerkiksi käyttöliittymäkomponenttikirjastoja. Tästä syystä Adobe PhoneGap:n

¹⁷ Nykyinen saatavilla oleva SAP UI5:n versio 1.4.3 (<http://scn.sap.com/community/developer-center/front-end>) ei vielä ole toteutettu mobiilikäyttöön.

lisäksi käytetäänkin usein jotain käyttöliittymäkirjastoa, kuten jQueryMobilea, joskaan tämä ei ole välttämätöntä.

Adobe PhoneGap ei sisällä valmiita toteutuksia sovelluksen käyttöliittymän rakentamiseksi. Se ei esimerkiksi sisällä käyttöliittymäkomponentteja tai tarjoa arkkitehtuuria, kuten MVC-arkkitehtuuri¹⁸, sovelluksen rakentamiseksi. Tästä syystä sen vertailu valmiita käyttöliittymäkomponentteja sisältäviin kirjastoihin, kuten Sencha Touch, ei kaikilta osin ole tarkoituksenmukaista. Se on silti valittu tutkimuksessa tarkasteltavaksi, koska se sisältyy SAP:n tukemiin avoimen lähdekoodin mobiilisovelluskehityskirjastoihin ja tästä syystä se on arvioitu varteenotettavaksi vaihtoehdoksi mobiilisovelluksen kehittämiseen SAP-ympäristössä.

7.3 Appcelerator: Titanium Development Platform

Appcelerator: Titanium Development Platform (tai Appcelerator Titanium Mobile ja lyhyemmin Appcelerator) on avoimen lähdekoodin sovelluskehys web-sovellusten, hybridisovellusten ja erityisesti natiivien mobiilisovellusten kehittämiseen.

Appceleratorilla sovelluskehitys tapahtuu JavaScript:llä, kuten esimerkiksi PhoneGap:ssä, mutta monista muista mobiilisovelluskehityksistä poiketen ohjelma käännetään ennen sen ajamista natiiviksi sovellukseksi, joskin sovelluksen JavaScript:llä toteutettu logiikka tulkitaan vasta ajon aikana. Sovellusta ei siis ajeta hybridi-web-näkymässä, kuten esimerkiksi PhoneGap, vaan se ajetaan suoraan natiivina sovelluksena. Appcelerator onkin syvemmin integroitu ajoalustaansa ja se käyttää esimerkiksi alustojen omia käyttöliittymäkomponentteja ja palveluita eroten näin hybridisovellusten tavasta, jossa käytetään esimerkiksi HTML5-komponentteja.

Appceleratorin tiukan ajoalustaintegraation ansiosta se on suorituskyvyltään lähellä natiiveja sovelluksia. Tiukka integraatio mahdollistaa myös alustakohtaisten toiminnallisuuksien hyödyntämisen. Joskin tämä tapahtuu siirrettävyyden kustannuksella. Appcelerator mahdollistaa myös alustariippuvaisen sovelluskoodin ajon, joten sillä voidaan esimerkiksi hyödyntää tietyn alustan toiminnallisuuksia, kuten Androidin valikko-näppäintä, sovelluksen ajoalustasta riippuen. Appcelerator tarjoaa myös toiminnallisuutta ja ohjeita sovelluksen käyttöliittymän toteuttamiseksi alustan mukaisella tavalla, mikä on tärkeää yhtenäisen käyttökokemuksen aikaansaamiseksi. Joka tapauksessa Appceleratoria käytettäessä tarvitaan aina jonkin verran alustakohtaista kehitystä käyttöliittymän ja toiminnallisuuden näkökulmasta. Appceleratorin dokumentaation perusteella koodin alustariippumattomuus on 80 prosentista. (Appcelerator Inc. 2012)

7.4 Sencha Touch

Sencha Touch (tai Sencha: Touch) on avoimen lähdekoodin¹⁹ JavaScript-kirjasto mobiilien HTML5-pohjaisten sovellusten luomiseen. Sencha Touch:ssa kaikki toiminnallisuus on Javascript-pohjaista ja toteutettu vain kahden HTML5-elementin avulla. Näin ollen se eroaa tavallisen web-sivun rakenteesta, jossa on useita HTML-elementtejä ja joita muokataan esimerkiksi jQueryMobilea käytettäessä elementtien järjestystä muuttamalla

¹⁸ MVC-arkkitehtuuri on ohjelmistoarkkitehtuuri, jossa käyttöliittymän toteutus on erotettu sovelluksen varsinaisesta käsittelylogiikasta. Se koostuu kolmesta osasta, jotka ovat: malli, näkymä ja käsittelijä.

¹⁹ Sencha Touch on avoimen lähdekoodin sovelluskehys. Sillä kehittäminen on kuitenkin maksullista kehitettäessä sulautettuja sovelluksia yli 5000 yksikön laitekannalle tai mikäli sen pohjalta toteutetaan kaupallista sovelluskehityskirjastoa tai mobiilikohitystyökalua.

(<http://www.sencha.com/products/touch/license/>)

JavaScript:llä. Sencha Touch:n rakentaminen kahden HTML5-elementin varaan tarkoittaa, että esimerkiksi tavalliset selaamisen toiminnot, kuten edelliselle sivulle siirtyminen ja linkittäminen tiettyyn osaan sovellusta, täytyy ottaa huomioon ja toteuttaa erikseen sovellusta kehitettäessä. Toisaalta, tämä ominaisuus on tyypillinen muillekin web-sovelluksille, joiden ei välttämättä ole tarkoituksaan toimia perinteisen web-selaamisen mallin mukaisesti. Tällöin käyttöliittymä voidaan tehdä myös enemmän mobiiliuden kuin web-selaamisen näkökulmasta. (Sanjeet ym. 2012)

Sencha Touch on suunniteltu natiivinkaltaisten web- ja hybridisovellusten kehittämiseen. Se mahdollistaa sekä sovellusten käytön HTML5-tuetuilla selaimilla että sovellusten pakkaamisen hybridisovellukseksi ja sovelluksen asentamisen sekä ajamisen Android- tai iOS-sovelluksena.

7.5 Sybase Mobile SDK

Sybase Mobile SDK on SAP:n tarjoama maksullinen sovelluskehitystyökalu hybridi- ja natiivimobiilisovellusten kehittämiseen. Se tarjoaa yhdenmukaisen mallin sekä välineitä, kuten uudelleen käytettäviä käyttöliittymäkomponentteja ja käyttäjän autentikoinnin toteutuksia, mobiilisovellusten kehittämiseksi kaikille tuetuille alustoille, joita ovat iOS iPhone:lle ja iPad:lle, Android, BlackBerry sekä Windows Mobile. Sybase Mobile SDK mahdollistaa myös alustakohtaisten ominaisuuksien ja toiminnallisuuksien hyödyntämisen tarjoamalla alustojen omista kehityspaketeista (SDK) sekä suunnitteluohjeistuksista laajennetut sovelluskehitystyökalut. Sybase Mobile SDK on kehitetty erityisesti yrityksen mobiilien tarpeiden, kuten käyttöoikeuksien hallinnan sekä yhteydettömässä tilassa tapahtuvan tiedon hallinnan, näkökulmasta. (Sanjeet 2012)

Sybase Mobile SDK -kehityspaketilla eli kehitysalustalla luodut sovellukset, jotka voidaan kehittää joko hybridisovelluksina yksinkertaisempiin tarkoituksiin tai natiiveina monimutkaisempiin sovelluksiin, ajetaan mobiililaitteelle asennettavalla Hybrid Web Container:lla. Hybrid Web Container on osa Sybase Unwired Platform –tuoteperhettä (SUP).

8 Aineiston käsittely ja tulokset

Tutkimuksen aineisto on kerätty pääosin sovelluskehysten dokumentaatioiden, sovellusesimerkkien sekä Internet-lähteistä kerättyjen sovelluskehysiin liittyvien havaintojen perusteella. Lisäksi tutkimuksessa on perehdytty sovelluskehysten asennuspaketteihin ja ohjelmakoodeihin tekemättä kuitenkaan varsinaista sovelluskehitystä. Aineiston kerääminen on toteutettu laatuominaisuuksien arviointikriteerien näkökulmasta. Arviointikriteerien ulkopuolisia huomioita on kerätty, mikäli ne on arvioitu hyödyllisiksi sovelluskehysten valintaongelman kannalta. Analyysiä tehdessä toiminnallisuuden ja siirrettävyyden kriteerit osoittautuivat tärkeimmiksi, joten niiden pohjalta tuloksia on tarkasteltu muita laatuominaisuuksia tarkemmin alalukuihin jaoteltuna. Arviointi on toteutettu tutkimuksen tekohetkellä (alkusyksy 2012) saatavilla olevien seuraavien versioiden perusteella:

SAP UI5 1.4.3

Adobe PhoneGap 2.1.0

Appcelerator: Titanium Development Platform 2.1

Sencha: Touch 2.0.1

Sybase Mobile SDK 2.1.

8.1 Sovelluskehysten toiminnallisuus-laatu

Toiminnallisuus ISO 9126:ssa määriteltynä laadun osatekijänä kuvaa ohjelmiston kykyä tarjota tarkoituksenmukaiset toiminnot sekä täyttää halutut tarpeet, kun ohjelmistotuotetta käytetään tietyissä olosuhteissa. Sovelluskehys arvioidessa toiminnallinen laatu voidaan nähdä olennaisimpana laadun tekijänä, koska sen avulla voidaan arvioida, miten hyvin sovelluskehys sisältää kehitettävässä sovelluksessa tarvittavaa toiminnallisuutta valmiina. Toiminnallista laatua voidaan käyttää sovelluskehysten valinnassa selvittämään, mitä toiminnallisuutta ne tarjoavat kontekstiin liittyen ja minkä toiminnallisuuden toteuttaminen vaatii lisätyötä tai ei ole mahdollista.

8.1.1 Soveltuvuus

Sovelluskehysten toiminnallisuus-laatua soveltuvuuden osalta arvioidaan sen perusteella, miten hyvin ne toteuttavat luvussa 4 esitettyjen toiminnallisuuksien toteuttamisen. SAP-toiminnallisuuksista Yhdistettävyyden SAP-dataan -luvun tekijöitä arvioidaan kuitenkin yhteensopivuus-alaominaisuuden analyysissä ja Käyttöoikeuksien hallinta-luvun tekijöitä arvioidaan turvallisuus-alaominaisuuden analyysissä. Muilta osin tärkeimpiä sovellusominaisuuksia arvioidaan soveltuvuus-alaominaisuuden analyysin osana.

Sovelluskehykset eroavat toisistaan tarjoamansa karttatoiminnallisuuden osalta. Sybase Mobile SDK mahdollistaa natiivien karttakomponenttien käyttämisen suoraan. Myös Appcelerator tarjoaa natiivit karttakomponentit käytettäväksi, mutta siinä niitä ei käytetä suoraan, vaan Titanium.Map-moduulin avulla. Myös Sencha Touch sisältää valmiin toteutuksen kartan integroimiseksi sovellukseen. Toteutus on rakennettu Google Maps:n

JavaScript-version päälle, joten Googlen Maps:n lisenssiehdot²⁰ pätevät sen käyttöön. Myös Sybase Mobile SDK:lla ja Appceleratorilla natiivikarttojen hyödyntäminen sitoo näiden lisenssiehtoihin. SAP UI5 ja PhoneGap eivät tarjoa ollenkaan valmista toteutusta kartan integroimiseksi sovelluksiin. Niillä, kuten tosin myös muilla sovelluskehyksillä, voidaan käyttää kolmannen osapuolen sovelluskirjastoja, joita ovat esimerkiksi leaflet²¹ ja OpenLayers²² JavaScriptille, route-me²³ ja MapBox²⁴ iOS:lle sekä osmdroid²⁵ Androidille.

Karttatoiminnallisuuteen liittyy kiinteästi myös käytettävät kartta-aineistot esimerkiksi niiden kattavuuden ja ajantasaisuuden näkökulmasta. Kartta-aineistot ladataan pääsääntöisesti Internetistä sovellusta käyttäessä, joskin poikkeuksiakin on. Esimerkiksi OsmDroid mahdollistaa kartta-aineistojen lataamisen laitteelle. Kolmannen osapuolen karttakirjastot mahdollistavat yleensä kartta-aineistojen vapaan käytön. Natiiveissa karttakomponenteissa taas ei käytettäviä kartta-aineistoja voi välttämättä vaihtaa ja se voi aiheuttaa ongelmia, jos tarpeena on käyttää omia kartta-aineistoja tai käytettävät kartta-aineistot ovat virheellisiä²⁶. Tästä syystä Sybase Mobile SDK:lla ja Appcelerator:lla on syytä tarkastella natiivien karttakomponenttien aineistojen riittävyyttä sovellustarpeen näkökulmasta ennen sovelluskehysten valintaa.

Ulkopuolisia kartta-aineistoja tai -palveluja käyttäessä myös niihin liittyvät käyttöehdot ovat merkittävä tekijä. Esimerkiksi pilvipalveluja sovelluksen tiedon tallentamiseen käytettäessä, pilvipalvelun tarjoaja voi saada tarpeettoman laajat oikeudet tallennettuun tietoon. Esimerkiksi Google:n Fusion Tables:iin tietoja tallentava paikkatietosovellus mahdollistaa Google:lle näiden tietojen hyödyntämisen tai niiden eteenpäin jakamisen tietyin ehdoin (Google 2011). Onkin tärkeää tutustua tarkasti karttatoiminnallisuuden ja kartta-aineistojen käyttöehtoihin sekä sopia tarpeenmukaisesta lisenssitasosta palvelun tarjoajan kanssa, ennen kartta-aineiston valintaa.

Kaikki sovelluskehykset mahdollistavat mobiililaitteen paikannusominaisuuksien hyödyntämisen. Sybase Mobile SDK ja Appcelerator tarjoavat mobiililaitteen natiivin paikannuksen ominaisuuksien, kuten paikannustarkkuuden asettamisen usealla eri vaihtoehdolla (vaihtoehtojen määrä riippuu alustasta), hyödyntämisen. Niillä on siis mahdollista hakea tarkka sijainti, vain kun se tarvitaan säästäten näin akun käyttöä. Myös SAP UI5, PhoneGap ja Sencha Touch mahdollistavat paikannustarkkuuden asettamisen, mutta vain kahdella eri vaihtoehdolla (enableHighAccuracy-parametrin avulla). Näiden sovelluskehysten paikannustoteutukset käyttävät selaimen HTML5-

²⁰ Google Maps:n lisenssiehdot. [Viitattu 31.10.2012]. Saatavissa:

<https://developers.google.com/maps/licensing>.

²¹ Leaflet on avoimen lähdekoodin JavaScript-kirjasto erityisesti mobiilikarttojen toteuttamiseen. [Viitattu 31.10.2012]. Saatavissa: <http://leafletjs.com/>.

²² OpenLayers on avoimen lähdekoodin JavaScript-kirjasto karttojen toteuttamiseen. [Viitattu 31.10.2012]. Saatavissa: <http://openlayers.org/>.

²³ Route-me on avoimen lähdekoodin sovelluskirjasto karttasovellusten tekemiseen iOS:lle. [Viitattu 31.10.2012]. Saatavissa: <https://github.com/route-me/route-me>.

²⁴ MapBox on sovelluskirjasto karttasovellusten tekemiseen esimerkiksi iOS:lle. [Viitattu 31.10.2012]. Saatavissa: <http://mapbox.com/>.

²⁵ Osmdroid on avoimen lähdekoodin sovelluskirjasto karttasovellusten tekemiseen Android:lle. [Viitattu 31.10.2012]. Saatavissa: <http://code.google.com/p/osmdroid/>.

²⁶ Kartta-aineistojen virheet voivat olla niin merkittäviä, että karttasovelluksen toteuttava yritys suosittelee käyttämään jotain toista karttasovellusta heidän sovelluksensa sijaan. Tästä esimerkkinä Apple:n 2012 julkaisema kartta-aineisto. [Viitattu 6.12.2012]. Saatavissa: <http://www.apple.com/letter-from-tim-cook-on-maps/>.

spesifikaatioluonnoksen paikannusominaisuutta²⁷, joka ei kuitenkaan edellytä selaintoteutuksia hyödyntämään paikannustarkkuuden syötettä paikannuksessa (W3C, 2012c).

Navigointi-toiminnallisuuteen liittyy sekä orientaation ja kompassisuunnan että sijainnin seuranta. Lisäksi myös reititys kuuluu olennaisesti navigointiin, mutta sen hyödyntäminen tutkimuksen sovelluskehyksillä edellyttää käytännössä ulkopuolisten sovelluskirjastojen ja palvelimien käyttöä. Laitteen orientaation ja kompassisuunnan selvittäminen sekä sijainnin seuranta on sovelluskehiksestä riippuen mahdollista joko laitteen omien rajapintojen avulla tai W3C:n laiteorientaatiokesifikaation(W3C, 2012b) ja sijaintikesifikaation(W3C, 2012c) määrittelemien rajapintojen avulla. Kaikki sovelluskehikset mahdollistavatkin orientaation sekä kompassisuunnan hyödyntämisen sovelluksissa edellyttäen, että mobiililaitteessa on tarvittavat fyysiset anturit. Lisäksi web-sovelluksia SAP UI5:llä ja Sencha Touch:lla toteutettaessa on edellytyksenä, että selain implementoi laiteorientaatiokesifikaation.

Paikkatietoanalyysijä sovelluskehikset tarjoavat vain hyvin rajallisesti, joten paikkatietoanalyysien kannalta sovelluskehiksiä vertailulla ei ole suurta merkitystä. Lähinnä tarjolla olevat paikkatietoanalyysit rajoittuvat vain geokoodaukseen tai vastakkaiseen geokoodaukseen, mutta niidenkin toteutukset hyödyntävät ulkopuolisia palvelimia, ainakin Android:n²⁸ ja iOS:n²⁹ osalta. Tästä syystä niitä saattavat rajoittaa ulkopuolisten palveluiden käyttöehdot, kuten kiello geokoodaustulosten tallentamiselle, tai rajoitteet, kuten maksimissaan tietty määrä pyyntöjä vuorokaudessa. Kaikki sovelluskehikset mahdollistavat ulkopuolisten paikkatietoanalyysipalveluiden hyödyntämisen, mutta niiden hyödyntäminen edellyttää käytännössä myös ulkopuolisten sovelluskirjastojen käyttöä. Näin ollen paikkatietoanalyysien kannalta ulkopuolisten sovelluskirjastojen valinta on huomattavasti merkitsevämpi tekijä, kuin mitä varsinaisen sovelluskehiksen valinta on.

Kaikki sovelluskehikset tukevat tiedon tallentamista, mutta eroavat tarjolla olevien toteutusten suhteen. PhoneGap, Appcelerator ja Sybase Mobile SDK mahdollistavat tiedon tallentamisen laitteen tiedostojärjestelmään ja sovellukselle dedikoituun tietokantaan. PhoneGap ja Appcelerator tarjoavat molemmille tallennustavoille laiteriippumattomat rajapinnat, kun taas Sybase Mobile SDK:lla tallennus voidaan toteuttaa laitteen omilla rajapinnoilla. SAP UI5 ja Sencha Touch sen sijaan tarjoavat vain selaimen rajapintojen mahdollistamat tiedon tallennustavat, kuten Web Storage ja Web SQL Database. SAP UI5 ei tarjoa näiden hyödyntämiseksi toteutusta, vaan tiedon tallentaminen edellyttää selaimen rajapintojen suoraa hyödyntämistä ja Sencha Touch:kin tarjoaa vain valmiin toteutuksen Web Storage:n hyödyntämiseksi. Näin ollen SAP UI5:llä ja Sencha Touch:lla tietoa selaimen rajapintojen avulla tallentaessa, on tarpeen varmistaa toteutuksen toimivuus eri selaimilla.

²⁷ Sencha Touch:lla on myös mahdollista hyödyntää natiivia paikannusominaisuutta sillä sovelluksia natiivipakatessa.

²⁸ Android Geocoder. [Viitattu 6.11.2012]. Saatavissa: <http://developer.android.com/reference/android/location/Geocoder.html>.

²⁹ iOS Geocoder. [Viitattu 6.11.2012]. Saatavissa: http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/LocationAwarenessPG/UsingGeocoders/UsingGeocoders.html#//apple_ref/doc/uid/TP40009497-CH4-SW5.

Tietojen synkronointiin sovelluskehysistä paras tuki on Sybase Mobile SDK:ssa. Sillä on mahdollista MBO:ta käyttäessä määritellä esimerkiksi miten usein ja millä tavoin mitkäkin tiedot synkronoidaan. Sybase Mobile SDK mahdollistaa myös synkronointikonfliktien hallintamenetelmän³⁰ määrittelyn. Myös Sencha Touch:iin on saatavilla lisäosana (Sencha.io, beta-asteella), joka tukee tiedon synkronointia. Sillä tieto kuitenkin synkronoidaan pilveen ja synkronointitoiminnallisuuden painopiste on laitteiden välisessä tiedon synkronoinnissa palvelimen ja laitteen välisen synkronoinnin sijaan. Se ei myöskään mahdollista synkronointikonfliktien hallintamenetelmän valintaa, vaan siinä uusin tieto korvaa aina vanhemman konfliktitilanteessa. Muut sovelluskehykset eivät sisällä synkronointitoiminnallisuutta, joten niillä joudutaan käyttämään kolmannen osapuolen toteutuksia, kuten mobeelizer³¹, tai kehittämään synkronointitoiminnallisuus itse.

8.1.2 Tarkkuus

Sovelluskehysten laatua tarkkuuden näkökulmasta on arvioitu niiden ensisijaisen tai ensisijaisten käyttökohteiden perusteella. Arviointi on tehty sen perusteella, onko sovelluskehykset suunniteltu ensisijaisesti mobiilisovellusten kehittämiseen, vai onko mobiilisovellusten kehittäminen vain yksi käyttökohde sovelluskehysille. Lisäksi sovelluskehysiksi on arvioitu niiden SAP-suuntautuneisuuden osalta. Onko sovelluskehykset kehitetty esimerkiksi ensisijaisesti SAP-sovellusten kehittämiseen vai onko SAP sovelluskehysille vain yksi mahdollinen käyttökohde, joka ei erotu edukseen muista käyttökohteista esimerkiksi käyttökohteistaisten toimintojen määrässä.

Kaikki muut arvioidut sovelluskehykset on suunniteltu ensisijaisesti mobiilisovellusten kehittämiseen lukuun ottamatta SAP UI5:ttä. SAP UI5:n käyttö on mahdollista myös mobiilik kehityksessä, mutta esimerkiksi kaikki sen sisältämät käyttöliittymäkomponentit eivät toimi mobiiliympäristössä. SAP UI5:lle on kuitenkin luvattu tuki mobiilikäytölle ja mobiilissa toimiville käyttöliittymäkomponenteille.

SAP:n tukemisen osalta sovelluskehykset jakautuvat kahteen luokkaan. Toisessa luokassa ovat SAP UI5 ja Sybase Mobile SDK, jotka on kehitetty erityisesti SAP-sovellusten tekoon³² ja jotka tukevat kattavasti muun muassa tietoturvan, saavutettavuuden ja SAP:iin yhdistämisen osalta SAP-standardeja (Ganz 2012; Sybase 2012). Toisessa luokassa ovat PhoneGap, Appcelerator ja Sencha Touch, jotka on kehitetty laajempaan käyttöympäristöön ja virallisesti yhteistyö SAP:n kanssa ja tuki SAP:n kanssa käytettäväksi on tullut alkuvuodesta 2012. Näiden kolmen sovelluskehysten osalta SAP on luvannut yhdessä sovelluskehysten kehittäjien kanssa tarjota kehittäjille avoimen lähdekoodin työkalut tehokkaaseen ja kustannuksia säästävään sovelluskehitykseen SAP-ympäristössä (SAP 2012b). Tuki sovelluskehysten SAP-tuen toteutuksille on kuitenkin vielä työn alla³³.

³⁰ Sybase Mobile SDK:lla synkronointikonfliktit voidaan määritellä ratkaistavaksi esimerkiksi aina uusimman tiedon automaattisesti valitsevalla tavalla tai käyttäjän valinnan edellyttämällä tavalla.

³¹ Mobeelizer on pilvipalvelu tiedon synkronoinnin toteuttamiseksi mobiilisovelluksiin. [Viitattu 16.11.2012]. Saatavissa: <http://www.mobeelizer.com/>.

³² SAP UI5:ttä ja Sybase Mobile SDK:ta on mahdollista käyttää SAP-sovellusten lisäksi muidenkin sovellusten toteuttamiseen.

³³ Valmiina on esimerkiksi tuki OData:lle Appceleratorille (Lisätietoa:

<http://www.appcelerator.com.s3.amazonaws.com/pdf/datasheet-SAP-odata.pdf> [Viitattu 26.10.2012]) sekä Sencha Touch:lle (Lisätietoa: <https://market.sencha.com/users/46/extensions/49> [Viitattu 26.10.2012]).

8.1.3 Yhteensopivuus

Sovelluskehysten laatu yhteensopivuuden osalta on arvioitu sekä SAP-järjestelmä- että paikkatietojärjestelmäintegroitavuuden kannalta. SAP-järjestelmän yhteyksien kannalta on tutkittu ennen kaikkea OData-yhteystapaa, koska se on SAP:n suosittelema ratkaisu mobiilisovellusten integroimiseksi SAP:iin.

SAP-järjestelmiin yhdistämisessä Sybase Mobile SDK erottuu edukseen mahdollistaen sekä OData:n että MBO:iden hyödyntämisen. MBO-ratkaisu mahdollistaa sovelluksen tietomallin määrittämisen palvelimella ja esimerkiksi tiedon synkronoinnin toteuttamisen automaattisesti ilman erillistä sovelluskohtaista kehitystä. MBO-ratkaisu tukee myös tiedon työntöä palvelimelta sovelluksille, mikä on varsinkin reaaliaikaista tietoa tarvitsevien sovellusten kannalta hyödyllistä.

Appcelerator ja Sencha Touch tukevat myös OData-palveluita mahdollistaen tiedon hakemisen sekä muokkaamisen palvelimilta. Myös SAP UI5:ssä on tuki OData:lle, joskin siinä tuki tiedon muokkaamiselle on vasta kehitysasteella eikä se ole virallisesti tuettuna. PhoneGap:ssä ei ole virallista tukea OData:lle ollenkaan, joskin OData-palveluiden käyttöön on mahdollista käyttää ulkopuolisia kirjastoja, kuten datajs³⁴.

Paikkatietojärjestelmiin integroimiseksi sovelluskehikset eivät varsinaisesti tarjoa valmiita ratkaisuja. Kaikki sovelluskehikset mahdollistavat kuitenkin ulkopuolisten sovelluskirjastojen hyödyntämisen paikkatietojärjestelmäintegraation tueksi. Ulkopuolisia kirjastoja tai niiden mahdollisia integraatiotapoja ei kuitenkaan sen tarkemmin ole arvioitu. Lähinnä eroja integroitavuudessa voi olla verratessa Sybase Mobile SDK:ta muihin sovelluskehiksiin, koska tällä sovelluskehitys tapahtuu yleensä natiiviohjelmakoodilla, kun taas muilla sovelluskehiksillä käytetään JavaScript-koodia, mikä vaikuttaa käytettävissä oleviin sovelluskirjastoihin.

8.1.4 Turvallisuus

Turvallisuutta arvioidaan ottaen huomioon kokonaisvaltaisesti, mitä ratkaisuja sovelluskehys tarjoaa tietoturvan huomioimiseksi sovelluksen tarvitseman tiedon osalta. Tietoturvan arviointi jaetaan sovelluskehiksittäin kolmeen osaan, jotka ovat: tietoturva tiedonsiirrossa, tietoturva laitteella sekä mahdollisuus hyödyntää SAP-ympäristössä yleisiä tietoturvaa parantavia ratkaisuja, kuten kertakirjautuminen (SSO).

Kaikki sovelluskehikset tukevat tietoturvaan tiedonsiirrossa mahdollistamalla salattujen SSL-yhteyksien käyttämisen. Sovelluskehyksistä PhoneGap ja Sencha Touch toimivat ainoastaan validien sertifikaattien kanssa. Appcelerator mahdollistaa sekä validien että epävalidien sertifikaattien hyödyntämisen tarpeen mukaan. Sybase Mobile SDK ja SAP UI5 eivät edellytä valideja sertifikaatteja. SAP UI5 antaa suojaa sivustojen välisten pyyntöjen väärentämiseltä (CSRF³⁵), kun tiedon siirto toteutetaan OData:lla. Muut sovelluskehikset eivät tältä implisiittisesti suojaa, joskin hybridi- tai natiivisovelluksia toteutettaessa CSRF ei ole ongelma, koska se liittyy tietoturvariskinä olennaisesti vain

³⁴ Datajs - JavaScript-kirjasto datakeskeisten sovelluksien toteuttamiseen. Datajs tukee muun muassa ODataa. [Viitattu 30.10.2012]. Saatavissa: <http://datajs.codeplex.com/>.

³⁵ Sivustojen välisen pyynnön väärentäminen (CSRF Cross-site request Forgery tai XSSRF) on tietoturvaongelma, jossa väärennetyjä komentoja lähetetään www-sivulta toiselle www-sivulle, joka tunnistaa käyttäjän kirjautumistietojen, kuten evästeiden avulla. CSRF:ää voidaan käyttää hyväksi esimerkiksi lataamalla kuvia tai JavaScriptiä keskustelupalstoille.

web-sovelluksiin. Tiedonsiirron tietoturvan tason suhteen sovelluskehykset eivät kuitenkaan eroa merkittävästi toisistaan.

Laitteella tapahtuvan tietoturvan osalta Sybase Mobile SDK erottuu selvästi muista sovelluskehyksistä. Se tukee lokaalin tiedon salaamista laitteella. Se mahdollistaa myös sovellusoikeuksien käyttäjäkohtaisen määrittämisen. Lokaalin tiedon salaaminen sekä sovellusoikeuksien määrittäminen antavat lisäturvaa esimerkiksi mobiililaitteen kadotessa tai joutuessa väärin käsiin eivätkä muut sovelluskehykset tarjoa kumpaankaan näistä käyttötapauksista valmiita ratkaisuja.

Kaikki sovelluskehykset mahdollistavat esimerkiksi kertakirjautumisen ja SAP:ssa määriteltävien käyttöoikeuksien hyödyntämisen, joten tämän osalta sovelluskehysillä kehitettyjen sovellusten tietoturvamahdollisuudet eivät eroa toisistaan. Sybase Mobile SDK:lla käyttöoikeuksia sekä kertakirjautumiskonfigurointeja voidaan määritellä SUP-palvelimella, kun taas muilla sovelluskehysillä kehitetyillä sovelluksilla voidaan käyttää esimerkiksi SAP Netweaver Gateway:tä. (Sanjeet ym. 2012)

Taulukko 8. SAP UI5:n laatu toiminnallisuuden näkökulmasta.

SAP UI5	
Soveltuvuus	<ul style="list-style-type: none"> - Edellyttää erillisen karttakirjaston käyttöä. - Orientaatio edellyttää laiteorientaatio-spesifikaatioluonnoksen (W3C 2012b) implementoivan selaimen. - Mahdollistaa HTML5:n tiedon tallennus ratkaisujen käytön, muttei tarjoa apua niiden käyttämiseksi.
Tarkkuus	<ul style="list-style-type: none"> + Kehitetty erityisesti SAP-sovellusten tekoon. - Kehitetty ensisijaisesti työpöytäsovellusnäkökulmasta. Mobiilioptimoitu versio tulossa.
Yhteensopivuus	<ul style="list-style-type: none"> + Tukee ODataa. - Tiedon muokkaaminen OData:lla vasta kokeiluasteella eikä virallisesti tuettua.
Turvallisuus	<ul style="list-style-type: none"> + Tukee SSL:n käyttöä. + ODataa käyttäessä suojaa CSRF:ltä. - Ei salattua tallennusta lokaalille tiedolle.

Taulukko 9. Adobe PhoneGap:n laatu toiminnallisuuden näkökulmasta.

Adobe PhoneGap	
Soveltuvuus	<ul style="list-style-type: none"> - Edellyttää erillisen karttakirjaston käyttöä. + Tiedon tallennus mahdollista laitteen tiedostojärjestelmään ja tietokantaan.
Tarkkuus	<ul style="list-style-type: none"> - SAP-tuki on ilmoitettu tulevan, mutta varsinaista toteutusta ei vielä ole.
Yhteensopivuus	<ul style="list-style-type: none"> - Vaatii kolmannen osapuolen toteutuksen OData-tukea varten.
Turvallisuus	<ul style="list-style-type: none"> + Tukee SSL:n käyttöä (edellyttää validia sertifikaattia). - Ei salattua tallennusta lokaalille tiedolle.

Taulukko 10. Appcelerator:n laatu toiminnallisuuden näkökulmasta.

Appcelerator: Titanium Development Platform	
Soveltuvuus	+ Tarjoaa natiivin karttakomponentin. + Mahdollistaa paikannuksen tarkkuuden määrittämisen. + Tiedon tallennus mahdollista laitteen tiedostojärjestelmään ja tietokantaan.
Tarkkuus	- SAP-tuki on ilmoitettu tulevan, mutta varsinaista toteutusta ei vielä ole.
Yhteensopivuus	+ Tukee ODataa.
Turvallisuus	+ Tukee SSL:n käyttöä. - Ei salattua tallennusta lokaalille tiedolle.

Taulukko 11. Sencha Touch:n laatu toiminnallisuuden näkökulmasta.

Sencha: Touch	
Soveltuvuus	+ Tarjoaa Googlen JavaScript-karttakomponentin. - Orientaatio edellyttää laiteorientaatio-spesifikaatioluonnoksen (W3C 2012b) implementoivan selaimen, jos kehitetään web-sovellusta. Tarjoaa myös rajapinnan laitteen orientaatorajapintojen hyödyntämiseen hybridisovelluksia kehitettäessä. - Mahdollistaa HTML5:n tiedon tallennus ratkaisujen käytön, mutta tarjoaa tukea vain Web Storage:n käyttöön. + Saatavilla synkronointituki pilveen (Sencha.io, beta-asteella)
Tarkkuus	- SAP-tuki on ilmoitettu tulevan, mutta varsinaista toteutusta ei vielä ole.
Yhteensopivuus	+ Tukee ODataa.
Turvallisuus	+ Tukee SSL:n käyttöä (edellyttää validia sertifikaattia). - Ei salattua tallennusta lokaalille tiedolle.

Taulukko 12. Sybase Mobile SDK:n laatu toiminnallisuuden näkökulmasta.

Sybase Mobile SDK	
Soveltuvuus	+ Tarjoaa natiivin karttakomponentin. + Mahdollistaa paikannuksen tarkkuuden määrittämisen. + Tiedon tallennus mahdollista laitteen tiedostojärjestelmään ja tietokantaan. + Paras tuki tietojen synkronoinnille.
Tarkkuus	+ Kehitetty erityisesti SAP-sovellusten tekoon.
Yhteensopivuus	+ Tukee ODataa + Tukee MBO:ita (Mahdollistaa esimerkiksi synkronoinnin ja muutoksien jälkeisen tiedon päivittämisen sovelluksille)
Turvallisuus	+ Kattavimmat mahdollisuudet tietoturvan huomioimiseen (SAP Afariaa käyttäessä vielä enemmän vaihtoehtoja). + Tukee SSL:n käyttöä. + Salattu tallennus lokaalille tiedolle. + Mahdollistaa käyttäjän sovellusoikeuksien, kuten sovelluksen avaamisen, rajaamisen.

8.2 Sovelluskehysten luotettavuus-laatu

Sovelluskehysten luotettavuudella tarkoitetaan niillä kehitettyjen sovellusten kykyä pysyä toiminnassa määritellyllä tavalla niitä määritellyllä tavalla käytettäessä. Sovelluskehyksillä luotettavuuteen vaikuttaa merkittävästi myös ajoalustan toteutus sovelluskehiksen toteutuksen lisäksi, koska niillä voidaan kehittää sovelluksia eri alustoille. Monille alustoille kehitettäessä eri alustoille kehitetyt sovellukset voivat myös toimia eri tavoin, joten luotettavuuden arviointi pelkästä sovelluskehysten näkökulmasta ei näin ollen ole täysin riittävää, vaan luotettavuuden analyysissä olisi hyvä ottaa huomioon myös eri alustat.

Virheiden siedon näkökulmasta sovelluskehikset eivät eroa toisistaan merkittävästi. Kaikki ne ovat testattuja, joten ne voitaneen olettaa sisäisesti hyvin toimiviksi. Lisäksi niiden sisäistä virheiden sietoa on vaikea arvioida luotettavasti ilman monipuolista testaamista. Virheiden välttäminen on kaikilla kehyksillä kehitettäessä sovelluskehittäjän vastuulla. Sovelluskehikset tarjoavat toisistaan eroavia sovellusarkkitehtuureja, kuten MVC-arkkitehtuuri, joilla muun muassa virhelähteitä voidaan vähentää sekä niiden vaikutusta tehdä paikallisemmaksi. Niitä ei kuitenkaan ole välttämättä suunnattu erityisesti virhelähteiden vähentämisen näkökulmasta. Eri arkkitehtuurien arvioiminen ei myöskään ole yksinkertainen ongelma (Babar ym. 2004), mistä johtuen sitä ei tässä tutkimuksessa tehdä.

Sovelluskehysten virheiden siedon näkökulmasta ainoa tutkimuksessa tehty eroja aiheuttava tekijä liittyy muistin vapautukseen eli roskien kerääjän toimintaan. Tutkimuksen sovelluskehyksillä roskien kerääjä toimii pitkälti automaattisesti ja muistin vapauttaminen ei ole sovelluskehittäjän vastuulla, joskin kehittäjän on kiinnitettävä huomiota siihen, että viittauksia ei jää käyttämättömiin objekteihin. Appcelerator on tässä kuitenkin poikkeus. Siinä kehittäjän on aktiivisesti muistettava nollata objektiivittaukset, jotta roskien kerääjä osaa vapauttaa objektien varaaman tilan muistista. Vaatimus objektiivittauksen

nollaamiselle johtuu siitä, että Appcelerator luo JavaScript-komentojen perusteella esimerkiksi natiiveja käyttöliittymäkomponentteja ja jos JavaScript:llä ei nollata objektiviittausta, ei natiivi komponentti poistu muistista.

Kaikki sovelluskehykset mahdollistavat virheiden sieppaamisen, mikä helpottaa virheiden vaikutuksista elpymistä. Virheiden sieppaamisella voidaan esimerkiksi yrittää toipua virheestä, raportoida käyttäjälle virheen syystä tai mahdollistaa tärkeiden tietojen tallentaminen, esimerkiksi lokaalisti tai palvelimelle, ennen sovelluksen kaatumista. Kehittäjän vastuulla on kuitenkin virrehallintaan liittyvien toiminnallisuuksien toteuttaminen kaikilla sovelluskehysillä. Lisäksi sieppaamattomien virheiden raportointi riippuu sovellusalustasta, joten tämän osalta sovelluskehysä ei voida vertailla.

Elpyvyyden kannalta merkittävä tekijä on mobiilien web-sovellusten toiminta virhetilanteissa. Mobiileissa web-sovelluksissa virheet saattavat jäädä havaitsematta, koska mobiiliselain ei välttämättä ilmoita ohjelmakoodin ajon aikaisista virheistä. Ajonaikainen virhe voi aiheuttaa sen, että ohjelma toimii suunnitellusta poikkeavalla tavalla, mutta käyttäjä ei välttämättä huomaa sitä. Vaihtoehtoisesti sovellus voi esimerkiksi kaatua ilman, että käyttäjä saa mitään virheilmoitusta. Ongelma koskee web-sovelluksissa sekä SAP UI5:ttä että Sencha Touch:ia. Myös Adobe PhoneGap:llä kehitetyt sovellukset ajetaan web-kehyksessä, joten ongelma koskee myös niitä. Appcelerator:lla ajon aikaiset sieppaamattomat virheet aiheuttavat ohjelman kaatumisen ja virheilmoituksen, jolloin ei ole vaaraa, että sovellus jäisi toimimaan halutusta poikkeavalla tavalla. Sybase Mobile SDK:lla toiminta virhetilanteen jälkeen riippuu sovellusalustasta. Natiivisovelluksia kehitettäessä virhetilanteiden hallinta on kuitenkin otettu web-sovelluksiin verrattuna paremmin huomioon esimerkiksi määritellyllä tavalla kaatumisen ja osittain myös kehittäjäohjeistuksien (Apple 2011) muodossa.

Taulukko 13. SAP UI5:n laatu luotettavuuden näkökulmasta.

SAP UI5	
Virheiden sieto	
Elpyvyys	- Ajonaikaisten virheiden hallinta alustan vastuulla. Käyttäjä ei välttämättä saa tietoa virheistä ja sovellus voi esimerkiksi jatkaa toimintaa epätarkoituksenmukaisella tavalla.

Taulukko 14. Adobe PhoneGap:n laatu luotettavuuden näkökulmasta.

Adobe PhoneGap	
Virheiden sieto	
Elpyvyys	- Ajonaikaisten virheiden hallinta alustan vastuulla. Käyttäjä ei välttämättä saa tietoa virheistä ja sovellus voi esimerkiksi jatkaa toimintaa epätarkoituksenmukaisella tavalla.

Taulukko 15. Appcelerator:n laatu luotettavuuden näkökulmasta.

Appcelerator: Titanium Development Platform	
Virheiden sieto	- Vaatii kehittäjältä aktiivista objektiivistausten nollaamista, jotta sovelluskehys merkitä objektit vapautettaviksi muistista. Jos objektiivittauksia ei nollata, on riskinä sovelluksen kaatuminen.
Elpyvyys	+ Kaatuu sieppaamattoman virheen jälkeen, mikä vähentää riskiä sovelluksen joutumisesta määrittelemättömään tilaan.

Taulukko 16. Sencha Touch:n laatu luotettavuuden näkökulmasta.

Sencha: Touch	
Virheiden sieto	
Elpyvyys	- Ajonaikaisten virheiden hallinta alustan vastuulla. Käyttäjä ei välttämättä saa tietoa virheistä ja sovellus voi esimerkiksi jatkaa toimintaa epätarkoituksenmukaisella tavalla.

Taulukko 17. Sybase Mobile SDK:n laatu luotettavuuden näkökulmasta.

Sybase Mobile SDK	
Virheiden sieto	
Elpyvyys	+ Kaatuu sieppaamattoman virheen jälkeen, mikä vähentää riskiä sovelluksen joutumisesta määrittelemättömään tilaan.

8.3 Sovelluskehysten käytettävyyks-laatu

Ymmärrettävyyttä on arvioitu sovelluskehysten mobiilisuuntatuneisuuden perusteella. Lisäksi on huomioitu keskitytäänkö sovelluskehysten dokumentoinnissa ensisijaisesti SAP- tai paikkatietosovelluksiin. Kaikki sovelluskehyykset on tarkoitettu ensisijaisesti mobiilisovellusten kehittämiseen SAP UI5:ttä³⁶ lukuun ottamatta ja tämä on myös havaittavissa sovelluskehysten dokumentoinnissa ja sovellusesimerkeissä, jotka keskittyvät mobiilikehitykseen ja huomioivat sen ominaispiirteet, kuten laitteen fyysisiin mittoihin mukautuvien käyttöliittymien tarpeen.

SAP UI5 ja Sybase Mobile SDK erottuvat sovelluskehyyksistä edukseen SAP-painottuneisuuden osalta. Molemmat on kehitetty erityisesti SAP-ympäristön sovelluskehitykseen ja niiden viralliset dokumentaatiot tarjoavatkin lukuisia esimerkkejä sovelluskehitykseen SAP-ympäristössä. Muita sovelluskehyyksiä ei sitä vastoin ole alkujaan kehitetty SAP-ympäristön sovellusten kehitykseen, ja niiden kehittäjät ovat tehneet virallista yhteistyötä SAP:n kanssa vasta joitain kuukausia. Tämä ilmenee myös

³⁶ SAP UI5:n 14.11.2012 julkaistu ennakkoversio 1.8 tarjoaa erityisesti mobiilisovellusten näkökulmasta kehitettyjä komponentteja tarjoten myös dokumentaatiota mobiilisovelluskehityksen tueksi. (Beck 2012)

sovelluskehysten dokumentoinneissa, joissa ei ole huomioitu SAP-ympäristöä millään tavoin erityisesti muista yritysjärjestelmäympäristöistä poiketen. Tämä saattaakin asettaa haasteensa PhoneGap:llä, Appcelerator:lla tai Sencha Touch:lla SAP-ympäristöön sovelluksia kehitettäessä, koska esimerkiksi moniin toiminnallisuustarpeisiin ei välttämättä ole saatavilla apua dokumentaatioista tai valmiista toteutuksista. SAP:n virallisen yhteistyön myötä lienee myös näillekin sovelluskehyksille tulevaisuudessa saatavilla paremmin ymmärrettäviä ja SAP:iin keskittyneitä dokumentaatioita.

Sovelluskehysten opittavuutta on arvioitu niille saatavilla olevien tukimuotojen perusteella, koska sovelluskehysten varsinaista opittavuutta on vaikeaa arvioida ilman kattavia testejä. Opittavuuden osalta sovelluskehukset eivät eroa merkittävästi toisistaan. Kaikille on saatavilla keskustelufoorumit, joilla on mahdollista esimerkiksi kysyä neuvoa ongelmatilanteissa. Kaikkien sovelluskehysten kehittäjät tarjoavat myös maksullisia tukipalveluita, joiden kautta on mahdollista saada eri tasoista tukea sähköpostituesta tietyllä vasteajalla palvelevaan tukeen.

Sovelluskehysten käytettävyysohjeistuksien mukaisuutta on arvioitu sen mukaan, huomioivatko sovelluskehukset mobiilialustojen käytettävyysohjeistuksia esimerkiksi mukauttamalla sovellusten käyttöliittymää näiden mukaisiksi, tarjoamalla kehysrakenteen alustakohtaisten käyttöliittymätoteutusten tekemiseen tai tarjoamalla ohjeita, miten eri alustojen käyttöliittymäohjeistukset huomioivia monialustasovelluksia voidaan kehittää sovelluskehyksellä.

Käytettävyysohjeistuksien huomioimisessa Sybase Mobile SDK ja Appcelerator erottuvat selkeästi parempina kehyskinä. Molemmilla huomioivat asennettavien sovellusten kehityksen osalta alustojen käyttöliittymäohjeistuksia kattavasti mukauttaen käyttöliittymää alustan mukaiseksi ilman erillistä kehitystä. Sybase Mobile SDK:ssa tämä on seurausta natiivilla ohjelmointikielellä kehittämisestä, mikä edellyttää erillisten alustakohtaisten toteutusten tekemistä eikä mahdollista kehysrakenteiden hyödyntämistä monialustakehityksessä. Appcelerator sen sijaan mahdollistaa saman ohjelmakoodin kääntämisen usealle eri alustoille sovelluksen käyttöliittymää näiden käyttöliittymäohjeistuksien perusteella mukauttaen. Esimerkiksi Appceleratorilla valintaikkuna mukautuu (katso Kuva 2) teemaltaan ja nappien sijainniltaan käytettävän alustan perusteella. Appcelerator tarjoaa myös kehysken, jolla voidaan toteuttaa alustakohtaista käyttöliittymätoiminnallisuutta jakamalla eri alustojen toteutukset erillisiin JavaScript-tiedostoihin näin selkeyttäen ohjelman rakennetta.

Myös Sencha Touch mahdollistaa alustaan mukautuvan käyttöliittymän toteuttamisen ja tarjoaa kehysrakenteen, laiteprofiilit (device profiles) siihen. Sencha Touch:lla kehitetyt sovellukset eivät kuitenkaan mukaudu käyttöliittymältään alustan perusteella automaattisesti, vaan mukautuminen vaatii aina erillistä kehitystä. Sencha Touch ei myöskään edes mahdollista kaikilta osin käyttöliittymäohjeistuksien mukaisten toteutusten tekemistä, koska sillä ei ole mahdollista hyödyntää sovelluksessa esimerkiksi fyysistä takaisin-nappia. Sencha Touch:n mukautumisessa on otettu lähtökohdaksi mukautuminen eri laitteiden toisistaan eroavien kokosuhteiden perusteella pyrkien tarjoamaan eri laitteilla yhtenevä käyttökokemus ja käytettävyys edellyttäen, että laitteiden mittasuhteet ovat toisiaan vastaavat, kun taas Appcelerator:lla lähtökohdana on tarjota laitteelle natiivina kehitettyjä sovelluksia vastaava käyttökokemus ja käytettävyys.

PhoneGap ja SAP UI5³⁷ eivät tue monialustakehitystä käyttöliittymäohjeistuksien hyödyntämisen näkökulmasta ollenkaan. Kumpikaan ei tarjoa kehysrakennetta monialustakehitykseen eikä kumpikaan tarjoa toteutusta, joka huomioisi sovellusalustan käyttöliittymäohjeistuksia käyttöliittymän tai käyttökokemuksen osalta. PhoneGap:llä tämä johtuu siitä, että se tarjoaa vain alustan hybridisovelluksien integroimiseksi mobiililaitteille eikä se tarjoa valmista toteutusta esimerkiksi sovelluksen rakenteelle tai käyttöliittymäelementeille. SAP UI5 sen sijaan on suunniteltu työpöytäkäyttöön, mikä on mobiilin käytettävyyden kannalta huono puoli, koska työpöytä- ja mobiilikäytössä käyttötarpeet voivat olla kovin erilaiset. Tästä syystä SAP UI5:ttä ei voida suositella käytettävyysohjeistuksien huomioimisen laadun tai ylipäänsä sovellusten käytettävyyden kannalta ollenkaan sovellusten kehittämiseen, mikäli halutaan toteuttaa sovelluksia pelkästään mobiililaitteelle.



Kuva 2. Appceleratorilla kehitetyn sovelluksen käyttöliittymän mukautuminen iPhoneen (vasemmalla) ja Androidin (oikealla) käyttöliittymäohjeistuksien mukaiseksi valintaikkunan osalta. (Appcelerator, 2012)

³⁷ SAP UI5:n kehitysversio 1.8.4 tarjoaa mobiilioptimoituja komponentteja, jotka huomioivat Androidin ja iPhoneen käyttöliittymäohjeistuksia esimerkiksi visuaalisen tyylin osalta.

Taulukko 18. SAP UI5:n laatu käytettävyyden näkökulmasta.

SAP UI5	
Ymmärrettävyys	- Dokumentaatiossa ei huomioida mobiilikehitystä juurikaan. + Painopiste SAP-kehityksessä.
Ohjeistuksien mukaisuus	- Ei huomioi käytettävyyttä mobiiliuden näkökulmasta. - Ei huomioi eri alustojen käytettävyysohjeistuksia eikä tarjoa ratkaisuja näiden tukemiseen

Taulukko 19. Adobe PhoneGap:n laatu käytettävyyden näkökulmasta.

Adobe PhoneGap	
Ymmärrettävyys	+ Painopiste mobiilikehityksessä.
Ohjeistuksien mukaisuus	- Ei huomioi eri alustojen käytettävyysohjeistuksia eikä tarjoa ratkaisuja näiden tukemiseen.

Taulukko 20. Appcelerator:n laatu käytettävyyden näkökulmasta.

Appcelerator: Titanium Development Platform	
Ymmärrettävyys	+ Painopiste mobiilikehityksessä.
Ohjeistuksien mukaisuus	+ Tarjoaa ohjeistuksen ja kehyksen miten kehitetään sekä Androidille että iPhonelle käytettävyydeltään alustan mukaisia sovelluksia.

Taulukko 21. Sencha Touch:n laatu käytettävyyden näkökulmasta.

Sencha: Touch	
Ymmärrettävyys	+ Painopiste mobiilikehityksessä.
Ohjeistuksien mukaisuus	+ Tarjoaa kehyksen alustakohtaisten erojen huomioimiseen kehityksessä. - Ei huomioi eri alustojen käytettävyysohjeistuksia.

Taulukko 22. Sybase Mobile SDK:n laatu käytettävyyden näkökulmasta.

Sybase Mobile SDK	
Ymmärrettävyys	+ Painopiste mobiilikehityksessä. + Painopiste SAP-kehityksessä.
Ohjeistuksien mukaisuus	+ Kehitys natiivikoodina tukee ohjeistuksien mukaista toteutusta. - Ei tarjoa kehystä alustakohtaisten erojen huomioimiseen kehityksessä.

8.4 Sovelluskehysten ylläpidettävyyys-laatu

Sovelluskehysten analysoitavuutta on arvioitu sovelluskoodin avoimuuden ja sovelluskehysten arkkitehtuurin dokumentoinnin perusteella. Kaikkien muiden sovelluskehysten paitsi Sybase Mobile SDK:n lähdekoodi on luettavissa selkokielisenä. Sybase Mobile SDK:n ohjelmakoodista on saatavilla vain konekielelle käännetty koodit, joten sen sisäisen toiminnallisuuden analysointi on muita sovelluskehyskiä vaikeampaa.

Jokaisen sovelluskehysten dokumentaatioissa kuvataan sen arkkitehtuuri yleisellä tasolla sekä kuvataan sovelluksen kehittämisprosessi yleisellä tasolla. Näin ollen sovelluskehysten välillä ei ole eroja arkkitehtuurin ja sovelluskehitysprosessin kuvaamisen tasolla, koska tutkimuksessa ei haluttu ottaa kantaa yksittäisten arkkitehtuurien välisiin eroihin.

Sovelluskehukset eroavat toisistaan kuitenkin ohjelmakoodiesimerkkien tarjoamisen osalta. Sencha Touch:n dokumentaatioissa komponenttien käytöstä on kuvattu selkeät esimerkit. Joidenkin komponenttien osalta dokumentaatioissa on myös videoita, joissa niiden toimintaa kuvataan sekä sovelluskoodin että käyttöliittymän näkökulmasta. Lisäksi joidenkin komponenttien toimintaa on mahdollista kokeilla interaktiivisten esikatselukehitystyökalujen avulla. Sencha Touch:n interaktiiviset esikatselukehitystyökalut mahdollistavat sovelluskoodin muokkaamisen reaaliaikaisesti. Sencha Touch mahdollistaakin hyvän sovelluskehysten sisäisen toiminnallisuuden ymmärtämisen useiden vaihtoehtoisten menetelmien avulla.

Appcelerator:n dokumentaatioissa komponenttien käyttämiseen on annettu selkeät esimerkit, jotka eivät kuitenkaan havainnollistavuudessa yllä Sencha Touch:n tasolle. SAP UI5:n dokumentaatioissa eri komponenteista on annettu lyhyet kuvaukset ja lisäksi komponenteista on tarjolla esikatselutyökalu, joka ei kuitenkaan ole interaktiivinen. Appcelerator:n ja SAP UI5:n dokumentaatiot tarjoavat molemmat kuitenkin käyttökelpoiset työkalut sovelluskehysten ymmärrettävyyden parantamiseksi. Sybase Mobile SDK ja PhoneGap eivät tarjoa muiden sovelluskehysten tavoin selkeitä esimerkkejä niiden käyttämiseksi tai sisäisen tarkemman toiminnan ymmärtämiseksi.

Muutettavuutta on arvioitu sovelluskehysten laajennusprosessin dokumentoinnin perusteella. Jokaisen sovelluskehysten dokumentaatioissa kuvataan yleisellä tasolla laajennusten tekemiseen läpikäytävän prosessin työvaiheet, joten tämän osalta sovelluskehukset eivät eroa toisistaan. Sovelluskehukset eroavat pienessä määrin laajennusdokumentointien osalta, mutta osiltaan erot johtuvat sovelluskehysten arkkitehtuurin eroista. Esimerkiksi PhoneGap:llä ja Appcelerator:lla laajennusesimerkit painottuvat mobiililaitteen toiminnallisuuksien hyödyntämiseen käytettävien lisäosien tekemiseen, kun taas SAP UI5:llä ja Sencha Touch:lla laajennusesimerkeissä kuvataan, miten uusia käyttöliittymäelementtejä voidaan toteuttaa. Sybase Mobile SDK:n dokumentaatioissa laajennusesimerkkejä ei erityisemmin tarjota, mikä johtuu sillä tehtävästä natiivisovelluskehityksestä, johon on saatavilla natiivialustojen omia laajennusdokumentointeja.

Sybase Mobile SDK:lla ja Appcelerator:lla sovelluksia voidaan testata virheitä etsimällä ja poistamalla³⁸ (debug) mobiililaitte-emulaattorilla. Sovellukset voidaan ajaa esimerkiksi

³⁸ Virheiden etsintä ja poistaminen (debugging) eroaa ohjelman tarkastamisesta (testing). Tarkastamisessa varmistetaan että ohjelma toimii määrittelyiden mukaisesti. Virheiden etsinnässä ja poistamisessa sitä vastoin pyritään löytämään ja korjaamaan ohjelmavirheiden lähteet.

Android-emulaattorissa, mikä mahdollistaa muun muassa verkkoyhteyksien häiriöiden simuloinnin tai GPS-koordinaattien asettamisen. Emulointi ei kuitenkaan korvaa täysin oikealla mobiililaitteella testaamista, koska se huomioi mobiiliuden erityispiirteet vain rajallisesti. Emuloimalla ei esimerkiksi välttämättä voida testata mobiilisovelluksia niiden oikeassa käyttöympäristössä.

Web-sovelluksia SAP UI5:llä, Adobe PhoneGap:llä ja Sencha Touch:lla kehitettäessä voidaan käyttää apuna työpöytäselaimia, joiden avulla ajonaikaisia virheitä voidaan etsiä. Työpöytäselaimella testaaminen ei kuitenkaan välttämättä riitä johtuen työpöytäalustan ja mobiilialustan eroista. Esimerkiksi mobiiliyhteyksien vaikutuksia sovelluksen toimintaan voi olla haastavaa testata työpöytäselaimessa, vaikka niiden vaikutuksia voitaisiinkin simuloida. Haasteita asettaa myös web-sovellusten mahdolliset toisistaan eroavat toiminnot eri mobiilialustoilla, mikä tosin on kaiken monialustakehityksen piirre.

Taulukko 23. SAP UI5:n laatu ylläpidettävyyden näkökulmasta.

SAP UI5	
Analysoitavuus	+ Lähdekoodi avointa. + Lyhyet kuvaukset eri komponenteista sekä esikatselutyökalu komponenttien tarkasteluun.
Testattavuus	- Sovellusten ajonaikaista toimintaa mobiililaitteilla on vaikea testata.

Taulukko 24. Adobe PhoneGap:n laatu ylläpidettävyyden näkökulmasta.

Adobe PhoneGap	
Analysoitavuus	+ Lähdekoodi avointa. - Ei tarjoa koodiesimerkkejä komponenttikehityksen tueksi.
Testattavuus	- Sovellusten ajonaikaista toimintaa mobiililaitteilla on vaikea testata.

Taulukko 25. Appcelerator:n laatu ylläpidettävyyden näkökulmasta.

Appcelerator: Titanium Development Platform	
Analysoitavuus	+ Lähdekoodi avointa. + Selkeät koodiesimerkit komponenttien käyttämiseksi.
Testattavuus	+ Mahdollistaa sovelluksen testaamisen mobiililaitetta simuloivan emulaattorin avulla.

Taulukko 26. Sencha Touch:n laatu ylläpidettävyyden näkökulmasta.

Sencha: Touch	
Analysoitavuus	+ Lähdekoodi avointa. + Dokumentaatioissa koodiesimerkkejä, videoita sekä interaktiivisia työkaluja sisäisen toiminnallisuuden ymmärtämiseksi.
Testattavuus	- Sovellusten ajonaikaista toimintaa mobiililaitteilla on vaikea testata.

Taulukko 27. Sybase Mobile SDK:n laatu ylläpidettävyyden näkökulmasta.

Sybase Mobile SDK	
Analysoitavuus	- Lähdekoodi ei saatavilla. - Ei tarjoa koodiesimerkkejä komponenttikehityksen tueksi.
Testattavuus	+ Mahdollistaa sovelluksen testaamisen mobiililaitetta simuloivan emulaattorin avulla.

8.5 Sovelluskehysten siirrettävyys-laatu

Siirrettävyys kuvaa ohjelmiston kykyä toimia ympäristöstä ja ympäristön tekijöistä riippumatta. Sovelluskehysten laatu siirrettävyyden näkökulmasta kuvaa esimerkiksi, kuinka monille eri alustoille sovelluskehyksillä voidaan kehittää sovelluksia sekä miten hyvin eri alustojen sovellukset osaavat hyödyntää alustojen yhteisiä tai alustakohtaisia ja toisistaan eroavia toiminnallisuuksia.

Siirrettävyyttä sopeutuvuuden, asennettavuuden ja korvattavuuden näkökulmasta on arvioitu ensisijaisesti sovelluskehysten dokumentaatioiden perusteella. Näistä on erityisesti tutkittu sovelluskehysten kehittäjien tuottamia dokumentaatioita sekä sovelluskehysten ohjelmointirajapintoja.

8.5.1 Sopeutuvuus

Sopeutuvuuden osalta sovelluskehysten laatua on arvioitu sovelluskehyksellä toteutettujen sovellusten kyvyllä tunnistaa ajoympäristönsä sekä kyvyllä mukautua ajoympäristöönsä. Kaikki sovelluskehykset SAP UI5:tä lukuun ottamatta osaavat tunnistaa mobiililaitteen sovelluksen ajon aikana. SAP UI5:llä mobiililaitteen tunnistaminen on myös mahdollista³⁹, mutta koska toteutus ei kuulu sovelluskehykseen, jää toteutuksen ylläpito sovelluskehittäjän vastuulle.

Alustakohtaisen mukautumisen osalta eri sovelluskehykset eroavat toisistaan merkittävämminkin kuin alustan tunnistamisen osalta. Appcelerator mukautuu parhaiten alustansa Sencha Touch:n mukautuessa lähes yhtä hyvin. PhoneGap ei taas mukaudu automaattisesti alustansa ollenkaan. Sybase Mobile SDK ja SAP UI5 sijoittuvat näiden väliin. Toisessa ääripäässä Appcelerator tarjoaa rajapinnat laitekohtaisiin ominaisuuksiin sekä mahdollistaa laiteriippuvaisten toteutusten tekemisen, kun taas PhoneGap mahdollistaa ainoastaan laitteen ominaisuuksien tunnistamisen - joskin tämän se tekee melko kattavasti - tarjoamatta ollenkaan ratkaisuja laitekohtaisten toteutusten tekemiseen.

Sekä Appcelerator⁴⁰ että Sencha Touch⁴¹ sisältävät selkeästi määritellyn kehysrakenteen alustasta riippuvaisten sovelluksen toiminnallisuuksien kehittämiseen. Kehysrakenne määrittelee tarkasti, miten alustakohtaiset sovelluksen osien eriävät toteutukset erotetaan toisistaan ilman, että alustariippumattomia ja yhteisiä toteutuksia tarvitsee toteuttaa useaan kertaan. Kehysrakenteen avulla esimerkiksi sovelluksessa olevan napin painamisen käynnistämä tiedon haku palvelimelta tarvitsee ohjelmoida vain kerran. Kehysrakenteiden avulla napin ulkoasu, sijainti tai tyyppi ei kuitenkaan tarvitse olla eri laitteilla sama, vaan esimerkiksi iPhone:lla se voi olla virtuaalinen nappi, jota painetaan kosketusnäytöltä, kun taas Androidissa se voi fyysinen laitteessa kiinteästi oleva nappi.

³⁹ SAP UI5 rakentuu jQuery:n päälle ja sillä voidaan tunnistaa mobiililaitteen tyyppi esimerkiksi linkissä esitetyllä tavalla. [Viitattu 19.10.2012]. Saatavissa: <http://www.jquery4u.com/mobile/detect-mobile-devices-jquery/#.UIEIk4bEKs0>.

⁴⁰ Ohje Appceleratorilla kehitettyjen sovellusten mukauttamiseen alustakohtaiseksi. [Viitattu 19.10.2012]. Saatavissa: http://docs.appcelerator.com/titanium/2.1/index.html#!/guide/Supporting_Multiple_Platforms_in_a_Single_Codebase.

⁴¹ Ohje Sencha Touch:lla kehitettyjen sovellusten mukauttamiseen alustakohtaiseksi. [Viitattu 19.10.2012]. Saatavissa: <http://docs.sencha.com/touch/2-0/#!/guide/profiles>.

Sybase Mobile SDK tarjoaa periaatteessa parhaan tuen laitekohtaisille toteutuksille, koska sillä sovellukset on mahdollista toteuttaa natiivisovelluksina. Sopeutuvuuden kannalta tämä ei kuitenkaan ole laadukasta, koska se vaatii eri alustakohtaisten sovellusten toteuttamisen lähes alusta alkaen. Ainoastaan jotkin kehyksen tarjoamat toiminnallisuudet, kuten tietoturvan ja tiedon hallinta, voidaan kehittää yhdellä kertaa eri alustoille. Sybase Mobile SDK mahdollistaa myös hybridisovellusten toteuttamisen, mutta tätä menetelmää ei kuitenkaan suositella kuin yksinkertaiset liiketoimintaprosessit, kuten hyväksynnit tai pyynnöt, toteuttavien sovellusten kehittämiseen (Sybase 2012). Myös SAP UI5 mahdollistaa selainalustan tunnistamisen ja se on suunniteltu myös monimutkaisempien sovellusten toteuttamiseen. Kummallaan sovelluskehysellä kehitetyt sovellukset eivät kuitenkaan tarjoa ollenkaan kehysrakennetta eikä juurikaan valmista toiminnallisuutta alustakohtaisten toteutusten tueksi ainakaan ilman lisäosia.

8.5.2 Asennettavuus

Sovelluskehysten laatua asennettavuuden kannalta on arvioitu sovelluskehyksille mahdollisten alustojen perusteella. Lisäksi asennettavuutta on arvioitu sen perusteella, miten hyvin eri alustojen toiminnallisuuksia sovelluskehys mahdollistavat suhteessa toisiin sovelluskehysiin.

Ehkä sovelluskehysten osalta merkittävin asennettavuus-laatuun vaikuttava tekijä on se, asennetaanko sovellukset mobiililaitteeseen natiivi- tai hybridisovelluksina vai ajetaanko ne laitteen selaimessa. SAP UI5:llä kehitetyt sovellukset ajetaan laitteen selaimessa. PhoneGap:llä sekä Sybase Mobile SDK:lla kehitetyt sovellukset taas edellyttävät asentamista. Mikäli sovelluskehys mahdollistaa molemmat tavat, lisää tämä sen monikäyttöisyyttä. Appcelerator ja Sencha Touch mahdollistavat sekä selaimessa ajettavien että asennettavien sovellusten tekemisen. Molemmat on kuitenkin kehitetty ensisijaisesti toiseen tarkoitukseen (Appcelerator hybridi- ja natiivisovelluksiin, Sencha Touch web-sovelluksiin), mistä johtuen niiden toissijaisen sovellustyyppin käytettävissä olevat toiminnallisuudet ovat rajatummalla. Myös mikäli Appcelerator:lla tai Sencha Touch:lla halutaan kehittää useita sovellustyyppisiä samanaikaisesti, vaatii se yleensä jonkin verran sovellustyyppikohtaista kehitystä eri sovellustyyppien mahdollistamien toiminnallisuuksien eroista johtuen.

Sovelluskehys erotetaan tukemiensa laitteiden ja alustojen perusteella. Käytännössä kaikilla selaimessa ajettavilla sovelluksilla tuottavilla sovelluskehyksillä vaatimuksena on HTML5-toiminnallisuuksia tukeva selain. Tämä on ongelma kuitenkin lähinnä vain, mikäli sovelluksia halutaan ajaa vanhempien mobiililaitteiden selaimilla. Nykyisten älypuhelimien selaimet toteuttavat jo usein sovelluskehysten vaatimalla tarkkuudella HTML5-toiminnallisuuksia. Sencha Touch on selainalustan suhteen vielä tarkempi edellyttäen WebKit-pohjaista⁴² selainta, mikäli halutaan olla varmoja sillä kehitettyjen sovellusten sovellustoiminnallisuuksien toimivuudesta. Etuna web-sovelluksia tuottavilla sovelluskehyksillä on, että ne eivät aseta tarkkoja vaatimuksia muun sovellusalustan, kuten mobiililaitteen tai käyttöjärjestelmän, suhteen.

⁴² WebKit on laajasti käytössä oleva vapaan lähdekoodin selainmoottori, joka on esimerkiksi Applen Safari- ja Googlen Chrome-selaimen pohjana. Lisäksi se on myös usean mobiilikäyttöjärjestelmän, kuten Apple iOS:n ja Androidin, oletusselaimena. [Viitattu 17.10.2012]. Saatavissa: <http://trac.webkit.org/wiki/Applications%20using%20WebKit>.

Hybridi- ja natiivisovelluskehikset taas asettavat tarkemmat vaatimukset soveltuvien alustojen suhteen. Tämä johtuu hybridi- ja natiivisovellusten tiukemmasta integraatiosta alustojen omiin toiminnallisuuksiin ja ominaisuuksiin sen sijaan, että tukeuduttaisiin kolmannen osapuolen toteutusmäärittelyihin, kuten HTML5. Taulukko 28 esittelee sovelluskehysten tuen eri alustoille asennettavien sovellusten kehityksen osalta.

Taulukko 28. Sovelluskehiksellä kehitettävien asennettavien natiivi- tai hybridi-sovellusten toimivuus eri alustoilla.

Sovelluskehitys-työkalu	iOS	Android	BlackBerry	WebOS	Windows Phone	Symbian	Bada
PhoneGap	X	X	X	X	X	X	X
Appcelerator	X	X	(X)				
Sencha Touch	X	X					
Sybase Mobile SDK	X	X	X		X		
SAP UI5	Ei mahdollista natiivi- eikä hybridisovellusten kehittämistä.						

PhoneGap:llä on paras tuki eri alustoille tutkimuksen hybridi- tai natiivisovelluskehyksistä. Se tukee alustoista iPhonea, Androidia, BlackBerryä OS:ää, WebOS:ää, Windows Phone 7:ää, Symbiania sekä Samsung Badaa (PhoneGap 2012). Sybase Mobile SDK:llä on myös melko laaja tuki eri mobiilialustoille. Se tukee seuraavia alustoja: iOS iPhone:lle ja iPad:lle, Android, BlackBerry ja Windows Mobile. Appcelerator ja Sencha Touch sen sijaan mahdollistavat ainoastaan iOS ja Android -sovelluskehityksen asennettavien sovellusten osalta, joskin Appceleratorilla on saatavilla Beta-versio⁴³ BlackBerry-kehitykseen. Varsinkin Appceleratorin suhteen tämä on merkittävä tekijä arvioitaessa sen käyttökelpoisuutta monialustaympäristössä, koska Appcelerator on kehitetty erityisesti asennettavien sovellusten tekoon.

Sovelluskehityksen mahdollistamat laitetoiminnallisuudet ovat olennaisia sovelluskehityksen valinnan kannalta. Esimerkiksi yhteydettömässä tilassa käytettävää sovellusta kehitettäessä on tärkeää, että sovelluksella voidaan tallentaa tietoja paikallisesti mobiililaitteelle. Toinen esimerkki on sijaintia hyödyntävä sovellus, jonka kehitystyökalun on syytä mahdollistaa sijaintitietojen selvittäminen puhelimen rajapintojen avulla. Tarpeelliset laitetoiminnallisuudet riippuvat kuitenkin olennaisesti kehitettävän sovelluksen tarpeista, joten sovelluskehiksiä arvioitaessa on tarpeen arvioida laajemmin, kuinka hyvin sovelluskehikset mahdollistavat toiminnallisuuksia verrattuna toisiin kehyksiin. Lisäksi oleellista on, miten sovelluskehys mahdollistaa laitekohtaisten toiminnallisuuksien hyödyntämisen eri laitteilla. Tutkituilla sovelluskehiksillä ei sen sijaan ole merkittävää eroa sen suhteen, miten ne mahdollistavat saman sovellustoiminnallisuuden eri alustoille, ellei sovellustoiminnallisuus ole ainoastaan tietylle laitteelle ominainen.

⁴³ Beta-versio BlackBerry-kehitykseen Appceleratorilla. [Viitattu 23.10.2012]. Saatavissa: <http://developer.appcelerator.com/doc/blackberry/getting-started-beta-preview>.

Sybase Mobile SDK mahdollistaa kattavimmin eri laiteominaisuuksien hyödyntämisen, koska siinä sovellukset kehitetään alustoille natiiveina. Yhteisten laiteominaisuuksien hyödyntäminen edellyttää kuitenkin erillisten toteutusten tekemistä eri laitteille. PhoneGap ja Appcelerator mahdollistavat myös hyvin laitteille yhteisten sekä laitekohtaisten toiminnallisuuksien hyödyntämisen. Näiden sovelluskehysten etuna Sybase Mobile SDK:hon verrattuna on myös, että ne eivät välttämättä edellytä alustoille yhteisten toiminnallisuuksien, kuten kameran tai tietoliikenneyhteyksien tilan, hyödyntämiseksi sovelluksessa laitekohtaista sovelluskehitystä. Sencha Touch ja SAP UI5 jäävät sen sijaan laitetoiminnallisuuksien hyödyntämisen osalta selvästi muista. Kummatkaan eivät web-tilassa - sen rajoitteista johtuen - mahdollista HTML5:een kuulumattomien laitetoiminnallisuuksien hyödyntämistä. Sencha Touch ei mahdollista myöskään natiivipakatuille sovelluksille web-sovellusominaisuuksien lisäksi kuin ainoastaan yhteyden tilan, järjestelmäilmoitusten, orientaation ja kameran hyödyntämisen hybridisovelluksille.

8.5.3 Korvattavuus

Korvattavuutta on analysoitu sovelluskehysten päivitettävyyden ja siihen liittyvien ohjeistuksien näkökulmasta. Sovellusten versiointi on myös osa päivitettävyyttä. Lisäksi päivitettävyyteen liittyy sovelluskehysten päivitettävyyden ja sen vaikutukset sovelluskehysten ohjelmarajapintoihin.

Sybase Mobile SDK:n kanssa käytettävä SUP-palvelin mahdollistaa sovellusten versioinnin monipuolisen hallitsemisen. SUP mahdollistaa esimerkiksi sovellusten uuden version päivittämisen etänä käyttäjän laitteelle, mikä ei ole muilla tutkimuksen sovelluskehyksillä mahdollista. Muista sovelluskehyksistä myös Appcelerator ja Sencha Touch mahdollistavat sovelluksen versioinnin, mutta käyttäjien sovellusten versiointiin ei ole mahdollista vaikuttaa etänä. Esimerkiksi Appcelerator:lla versioinnin tarkoituksena on mahdollistaa sovellusten päivitettävyys Android Marketin⁴⁴ kautta. Android Market:n hyödyntäminen yrityskohtaisten sovellusten jakelussa ei ole välttämättä järkevää, koska siellä sovellukset näkyvät kaikille ja yrityssovelluksia ei useinkaan ole tarkoitettu yleiseen jakeluun. SAP UI5 ja Adobe PhoneGap eivät sen sijaan tarjoa ollenkaan versiointitoteutusta niillä kehitetyille sovelluksille.

⁴⁴ Android Market on Android-puhelimissa oleva sovellus, jolla voidaan ladata uusia sovelluksia ja päivittää aiemmin ladattuja sovelluksia uusiin versioihin. [Viitattu 4.12.2012]. Saatavissa: <https://play.google.com/store/apps/details?id=com.google.android.finsky>.

Taulukko 29. SAP UI5:n laatu siirrettävyyden näkökulmasta.

SAP UI5	
Sopeutuvuus	<ul style="list-style-type: none">- Ei toteutusta ajoympäristön tunnistamiselle selainta lukuun ottamatta.- Ei huomioi mobiililaitetta muuten kuin selainikkunan koon osalta.
Asennettavuus	<ul style="list-style-type: none">+ Ei rajoitteita laitteen suhteen.- Tukee kunnolla vain HTML5-teknologioita implementoivia selaimia.- Ei mahdollista natiivi- tai hybridipakkaamista eli on ajettavissa vain selaimessa.- Kaikki kehyksen toiminnallisuudet eivät ole mobiilisovelluksissa mahdollisia
Korvattavuus	<ul style="list-style-type: none">- Ei sisällä versiointia.

Taulukko 30. Adobe PhoneGap:n laatu siirrettävyyden näkökulmasta.

Adobe PhoneGap	
Sopeutuvuus	<ul style="list-style-type: none">+ Mahdollistaa kattavasti ajoympäristön tunnistamisen.- Vaatii erillisen toteutuksen laitekohtaiseen mukautumiseen.
Asennettavuus	<ul style="list-style-type: none">+ Kehyksistä paras tuki eri laitealustoille.+ Mahdollistaa hybridikehityскеhykseksi paljon eri alustatoiminnallisuuksia. Lisäosilla (plugin) mahdollista toteuttaa laitekohtaisia toiminnallisuuksia.
Korvattavuus	<ul style="list-style-type: none">- Ei sisällä versiointia sovellusten näkökulmasta.

Taulukko 31. Appcelerator:n laatu siirrettävyyden näkökulmasta.

Appcelerator: Titanium Development Platform	
Sopeutuvuus	<ul style="list-style-type: none">+ Mahdollistaa kattavasti ajoympäristön tunnistamisen.+ Mahdollistaa alustakohtaisen sovelluskoodin ja tarjoaa menetelmän sovelluksen mukauttamiseen alustan mukaan.
Asennettavuus	<ul style="list-style-type: none">+ Mahdollistaa lähes natiivisovelluksia vastaavan laitteiden ominaisuuksien hyödyntämisen (myös laitekohtaisesti). Moduuleilla (module) mahdollista lisätä alustatoiminnallisuuksia.- Tukee laitealustoista ainoastaan Androidia ja iOS:ia (BlackBerryille saatavilla beta).
Korvattavuus	<ul style="list-style-type: none">+ Versiointi kehyksellä kehitetyille sovellukselle.

Taulukko 32. Sencha Touch:n laatu siirrettävyyden näkökulmasta.

Sencha: Touch	
Sopeutuvuus	+ Mahdollistaa kattavasti ajoympäristön tunnistamisen. + Mahdollistaa alustakohtaisen sovelluskoodin ja tarjoaa menetelmän sovelluksen mukauttamiseen alustan mukaan.
Asennettavuus	+ Ei suoraan rajoitteita laitteen suhteen. - Tukee kuitenkin kunnolla vain WebKit-pohjaisia HTML5-tekniologioita implementoivia selaimia. - Mahdollistaa natiivipakkauksen vain iPhone:lle ja Androidille. - Ei mahdollista mobiililaitteen ominaisuuksien hyödyntämistä, ellei natiivipakattu. Natiivipakattukin mahdollistaa vain yksittäisiä ominaisuuksia.
Korvattavuus	+ Versiointi kehyksellä kehitetyille sovellukselle.

Taulukko 33. Sybase Mobile SDK:n laatu siirrettävyyden näkökulmasta.

Sybase Mobile SDK	
Sopeutuvuus	+ Mahdollistaa kattavasti ajoympäristön tunnistamisen. - Ei mukaudu alustan perusteella ilman erillistä toteutusta.
Asennettavuus	+ Laaja tuki eri alustoille. + Mahdollistaa natiivisovelluksia vastaavan laitetoiminnallisuuden hyödyntämisen (vaatii alustakohtaiset toteutukset). - Laajempien sovellusten toteuttaminen vaatii käytännössä joko alustakohtaisten sovellusten tekemistä natiiveina tai ulkopuolisten sovelluskehysten käyttöä.
Korvattavuus	+ SUP-palvelimen avulla mahdollista sovellusten hallinta, kuten uusien versioiden asentaminen etänä.

9 Tulosten tarkastelu

Tutkimuksessa tarkasteltiin SAP:n suosittelemien sovelluskehysten laatua mobiiliin yritys ympäristön paikkatietosovellusten kehittämisen näkökulmasta. Tärkeimpänä tavoitteena oli selvittää, mikä sovelluskehyksistä soveltuu parhaiten edellä mainittujen sovellusten kehittämiseen. Luvussa 8 esitettyjen tulosten perusteella yhteenvetona voidaan sanoa, että Sybase Mobile SDK on tietyin reunaehdoin sovelluskehyksistä paras. Ero muihin sovelluskehysiin ei kuitenkaan ole suuri. Tässä luvussa tarkastellaan, miten sovelluskehykset eroavat toisistaan laatuominaisuuksien ja näiden alaominaisuuksien osalta.

Ehkä merkittävin tutkimuksessa tehty havainto on, että SAP UI5 soveltuu tällä hetkellä selkeästi muita tutkimuksessa arvioituja sovelluskehysia huonommin mobiilisovellusten kehittämiseen. Se on kehitetty ensisijaisesti työpöytäsovellusten toteuttamiseen. Lisäksi vaikka sillä kehitetyt sovellukset toimisivat osittain tai kokonaan mobiililaitteilla, ei sitä ole kehitetty mobiilikäytön näkökulmasta. Kaikki sen sisältämät käyttöliittymäkomponentit eivät esimerkiksi toimi mobiililaitteilla, eikä toimiviakaan komponentteja ole kehitetty mobiiliuden erityistarpeiden näkökulmasta.

Toisaalta SAP UI5:n etuna on, että se on erityisesti SAP käyttöön kehitetty, mikä erottaa sen positiivisesti Sybase Mobile SDK:ta lukuun ottamatta muista tutkimuksen sovelluskehyksistä. SAP UI5:lle on myös luvattu virallinen tuki mobiilisovellusten kehitykseen, joka löytyy jo esikatseluversiosta 1.8.4. Mobiiliuden näkökulmasta optimoidun version myötä myös SAP UI5:stä voi tulla varteenotettava vaihtoehto muille sovelluskehyksille.

SAP UI5 on hyvä esimerkki mobiilisovelluskehysten kehitysnopeudesta, mikä asettaa haasteensa sovelluskehysten valinnalle. Sovelluskehystä valitessa täytyy usein huomioida pidemmän ajanjakson sovellustarpeet (Lehman 1980; Kraftig ym. 2004). Sovelluskehykset kehittyvät kuitenkin niin nopeasti, että on vaikea ennustaa, millaisia ne ovat tai mitä toiminnallisuuksia ne tarjoavat lyhyemmän ajanjakson, kuten parin vuoden, päästä. Tämän perusteella liiallinen tukeutuminen yhteen tiettyyn sovelluskehykseen voi olla pitkällä aikajänteellä riskialtista ja se voi haitata jonkin toisen paremman sovelluskehysten hyödyntämistä jatkossa. Toisaalta tietyn sovelluskehysten valinta selkeyttää sovellusarkkitehtuuria ja mahdollistaa sovelluksesta riippumattomien toiminnallisuuden helpomman hyödyntämisen useammissa eri sovelluksissa. SAP:n kannalta onkin ymmärrettävää, että se ei tässä tilanteessa tukeudu vain yhteen sovelluskehykseen, vaan se tarjoaa useita eri vaihtoehtoja. Sovelluskehittäjän kannalta valinta on kuitenkin tehtävä.

Eri sovelluskehykset soveltuvat paremmin erilaisiin tarpeisiin, mikä tukee useamman sovelluskehysten rinnakkaista käyttöä yritys ympäristössä. Sovelluskehysia on myös mahdollista käyttää yhdessä. Esimerkiksi PhoneGap:ia voidaan käyttää SUP:ssa Hybrid Web Container:n sijaan hybridisovelluksia kehitettäessä (Sybase 2012). Lisäksi muun muassa Sencha Touch:ia voidaan käyttää yhdessä PhoneGap:n kanssa mahdollistaen hybridisovelluskehityksen ja mobiililaitteen natiivien ominaisuuksien hyödyntämisen. Yhdistelemällä eri sovelluskehysia onkin mahdollista jossain tapauksissa hyödyntää eri sovelluskehysten hyvät ominaisuudet ja välttää negatiiviset. Sovelluskehysten tarkastelu ja arviointi pelkästään toisistaan erillisenä ei anna kokonaiskuvaa ja sen saamiseksi olisikin syytä tarkastella myös erilaisia sovelluskehysten yhdistelmiä.

9.1 Tulokset laatuominaisuuksien näkökulmasta

Toiminnallisuus-laatu ja siirrettävyys-laatu ovat ehkä merkittävimmät laatukriteerit tutkimuksen tulosten kannalta. Lyhyesti tiivistettynä näistä ensimmäinen kertoo, miten paljon sovelluskehys tarjoaa valmista toiminnallisuutta tutkimuksessa tarkasteltavan mobiilikontekstin sovellusten toteuttamiseksi. Jälkimmäinen taas kuvaa, miten sovelluskehukset huomioivat eri sovellusalustat ja niiden tarjoamat mahdollisuudet, mikä on monia erilaisia alustoja sisältävässä mobiiliympäristössä tärkeää. Muiden laatuominaisuuksien osalta erot sovelluskehysten välillä eivät ole yhtä merkityksellisiä tai eroja on vaikea todentaa ilman yksityiskohtaista esimerkiksi suorituskykymittausten avulla tapahtuvaa analyysiä. Esimerkiksi käytettävyyden osalta sovelluskehysten käytettävyyttä laatuun tärkeämpää on, että loppukäyttäjille tehdyt sovellukset ovat käytettävyydeltään hyviä. Myös luotettavuus ja tehokkuus ovat laajemmalti merkittäviä kehitettyjen sovellusten kuin sovelluskehysten kannalta, vaikka sovelluskehysten toteutus vaikuttaakin jossain määrin sovellusten laatuun näiden ominaisuuksien osalta.

Toiminnallisuus-laadun kannalta Sybase Mobile SDK on sovelluskehyksistä paras. Se tarjoaa erityisesti yritys ympäristön sovellusten toteuttamiseen paljon valmiita ja tarpeellisia ominaisuuksia ja toimintoja, kuten tiedon tallentamismahdollisuuden laitteelle usein eri tavoin, laitteelle tallennettavan tiedon salauksen, laitteen ja palvelimen välisen tiedon synkronoinnin sekä sovelluskäyttöoikeuksien määrittämisen. Etuna Sybase Mobile SDK:lla on myös, että se on kehitetty erityisesti SAP-ympäristön sovellusten toteuttamiseen, mikä erottaa sen muista sovelluskehyksistä SAP UI5:ttä lukuun ottamatta.

Tarkasteltaessa sovelluskehysten toiminnallisuus-laatu yritysovelluskontekstin tarpeiden näkökulmasta, SAP UI5, Appcelerator ja Sencha Touch eivät eroa toisistaan merkittävästi. Jokainen näistä on jollain osa-alueella toisia parempi, mutta erot eivät ole niin selkeitä, että jokin näistä kolmesta erottuisi selkeästi muista. SAP UI5 tarjoaa esimerkiksi näistä parhaan tuen SAP-ympäristössä käytettäväksi. Appcelerator taas mahdollistaa parhaimman mobiililaitteen ominaisuuksien hyödyntämisen. Sencha Touch:lle taas on esimerkiksi saatavilla tiedon synkronointituki, jota SAP UI5:ssa tai Appcelerator:ssa ei ole. Kaikki nämä kolme sovelluskehystä jäävät kuitenkin jälkeen Sybase Mobile SDK:sta.

PhoneGap on toiminnallisuus-laadun näkökulmasta sovelluskehyksistä heikoin. Se on esimerkiksi sovelluskehyksistä ainoa, jossa ei ole tukea SAP:n suosittelemalle OData-tiedonsiirtoprotokollalle. PhoneGap:llä kehitettäessä voidaan kuitenkin hyödyntää muita sovelluskehysjä, kuten Sencha Touch:ia, jolloin sen heikot puolet voidaan osittain kiertää. Kuitenkin, kuten aikaisemmin on mainittu, tutkimuksessa oli tavoitteena vertailla sovelluskehysjä toisistaan irrallisina, joten sovelluskehysten yhteiskäyttöä ei tässä analysoida. Se on kuitenkin vartenotettava vaihtoehto jatkotutkimukselle.

Tutkimuksessa paikkatieto ja siihen liittyvät tekijät on toinen tärkeä osa kontekstia yritys ympäristön tekijöiden lisäksi. Paikkatietosovelluksen tarpeiden näkökulmasta sovelluskehukset eivät eroa toisistaan yhtä merkittävästi kuin yritysovellustarpeiden osalta. Erityisesti raskaampien paikkatietoanalyysijä hyödyntävien sovellustarpeiden osalta eroja sovelluskehysten välillä ei juuri ole, vaan tällaisia ominaisuuksia tarvitsevien sovellusten kehitys edellyttää kaikilla sovelluskehysillä ulkopuolisten sovelluskirjastojen käyttöä.

Kun tarkastellaan yleisiä paikkatiedon sovellusominaisuuksia, karttaa, paikannusta ja navigointia, eroja on kuitenkin havaittavissa. Näiden tarpeiden osalta natiivisovelluskehitykseen tarkoitettut Appcelerator ja Sybase Mobile SDK erottuvat selkeimmin muista sovelluskehityksistä. Niillä voidaan hyödyntää natiiveja karttakomponentteja ja niiden kartta-aineistoja, joskin lisenssiehtojen puitteissa. Ne mahdollistavat myös esimerkiksi paikannuksessa käytettävän tarkkuuden määrittämisen. Niillä voidaan hyödyntää myös mobiililaitteen antureita navigoinnin tukena esimerkiksi laitteen suuntimatietoja käyttäen. Sencha Touch tarjoaa lähes yhtä hyvät toiminnallisuudet kartan, paikannuksen ja orientaation osalta, joskin edellytyksenä on, että ajoympäristön selain tarjoaa rajapinnat näiden hyödyntämiseksi. SAP UI5 ja PhoneGap tarjoavat vähiten valmista toiminnallisuutta paikkatietotarpeiden näkökulmasta. Esimerkiksi kartan näyttäminen sovelluksessa edellyttää niillä kokonaan ulkopuolisten sovelluskirjastojen käyttämistä.

Siirrettävyyden osalta ehkä tärkein ominaisuus on tuki mahdollisimman monelle yleisesti käytössä olevalle alustalle sekä niiden tarjoamille toiminnallisuuksille. Nämä liittyvät asennettavuuteen, jonka osalta PhoneGap on tutkimuksessa arvioituista sovelluskehityksistä kokonaisuutena paras. Tuki useille eri alustoille on erityisen tärkeää yritysten laiteympäristön muutoksessa pelkkiä yrityksen tarjoamia laitteita sisältävästi laitteista myös käyttäjien henkilökohtaisia laitteita sisältäviksi. Käyttäjien henkilökohtaiset laitteet ovat usein moninaisempia merkeiltään ja malleiltaan, joten yrityksen on tarjottava aiempaa vähemmän laitesidonnaisia sovelluksia, jotta laitteita voidaan käyttää työssä. Henkilökohtaisten mobiililaitteiden käytöstä työnteon tukena käytetään nimitystä tuo-omalaiteesi (bring your own device, BYOD). (Bradley ym. 2012; Thomson 2012)

Natiivisovelluskehitykseen Appcelerator ja Sybase Mobile SDK tarjoavat hieman PhoneGap:iä paremmin mobiililaitteiden toiminnallisuuksia. Ero ei kuitenkaan ole merkittävä, koska PhoneGap:lle on saatavilla laajasti erilaisia lisäosia. Asennettavuuden toinen osatekijä, eli tuki eri laitealustoille, on Appcelerator:lla ja Sybase Mobile SDK:lla selvästi PhoneGap:iä heikompi ja se merkitseekin sovelluskehityksen valinnan kannalta enemmän. Erityisesti huomioitavaa on myös, että Appcelerator tarjoaa tuen vain iOS:lle ja Androidille. Toisaalta taas Appcelerator mahdollistaa Sybase Mobile SDK:sta poiketen samanaikaisen natiivisovelluskehityksen usealle eri alustalle, kun taas Sybase Mobile SDK:lla joudutaan kehittämään eri alustoille kokonaan omat sovelluksensa. Monialustakehityksessä Sybase Mobile SDK:lla kehityskulut voivatkin nousta huomattavasti suuremmaksi, koska jokainen sovellus täytyy toteuttaa erikseen

SAP UI5 ja Sencha Touch ovat asennettavuuden kannalta muita sovelluskehityksiä heikompia. Erityisesti mobiililaitteen toiminnallisuuksien hyödyntämisen kannalta ne eivät tarjoa hyvää tukea. Toisaalta niillä kehitettäessä sovellus voidaan kehittää täysin alustariippumattomana, jolloin sitä voidaan käyttää mahdollisesti myös tulevaisuuden mobiiliselaimilla. Edellytyksenä tosin on tuki HTML5:tä tukevalle (SAP UI5) tai WebKit:in (Sencha Touch) päälle rakennetulle selaimelle.

Asennettavuuden lisäksi sovelluskehityksen siirrettävyyteen vaikuttaa olennaisesti sen sopeutuvuus. Sopeutuvuuden kannalta ehkä tärkein ominaisuus sovelluskehityksillä on, että ne mahdollistavat ajoympäristön kattavan tunnistamisen, mikä on edellytys ajoympäristön natiivien toiminnallisuuksien hyödyntämiselle tai ajoympäristöön mukautumiselle. Ajoympäristön tunnistamisen kannalta SAP UI5 erottuu selkeästi negatiivisesti joukosta. Appcelerator ja Sencha Touch toisaalta mahdollistavat kattavan ajoympäristön

tunnistamisen ja tarjoavat myös välineet sovelluksen toiminnan mukauttamiselle ajoympäristön mukaiseksi. Appcelerator mukauttaa tämän lisäksi vielä sovelluksen käyttöliittymää alustan käytettävyysohjeistuksien mukaiseksi, mikä tekee siitä vielä paremman. Sopeutuvuuden ja käytettävyysohjeistuksien mukaisuuden linkittyminen toisiinsa on hyvä esimerkki siitä, että laatuominaisuudet liittyvät toisiinsa.

Toiminnallisuus- ja siirrettävyys-laadun kriteerien lisäksi ehkä kaksi merkittävintä kriteeriä tutkimuksessa olivat ymmärrettävyyden kriteeri sovelluskehityksen käyttökohdepainotuksista sekä analysoitavuuden kriteeri sovelluskehityksen lähdekoodin avoimuudesta. Sybase Mobile SDK on suunniteltu erityisesti mobiiliin SAP-käyttöön, joten se on käyttökohdepainotuksen puolesta paras sovelluskehys. SAP UI5 on myös kehitetty erityisesti SAP-sovellusten kehittämiseen, mutta sille ei kuitenkaan tutkimuksen tekohetkellä ollut tarjolla työkaluja mobiilisovelluskehitykseen. PhoneGap, Appcelerator ja Sencha Touch ovat sen sijaan mobiilikehitykseen suunniteltuja, mutta niitä ei ole kehitetty ensisijaisesti SAP-tarpeiden näkökulmasta, mikä voidaan nähdä negatiivisena puolena. Lähdekoodin avoimuuden osalta taas Sybase Mobile SDK jää jälkeen muista sovelluskehityksistä, koska se on ainoa sovelluskehys, jonka lähdekoodi ei ole luettavissa selkokielisenä. Lähdekoodin avoimuuden puute asettaa erityisvaatimukset laadukkaalle sovelluskehityksen dokumentoinnille.

Kokonaisuutena Sybase Mobile SDK voidaan arvioida parhaaksi sovelluskehitykseksi tutkimuksessa tarkasteltujen laatuominaisuuksien kannalta, mutta vain edellyttäen, että sovelluksen tarvitsee toimia vain yhdellä mobiilialustalla. Ero muihin sovelluskehityksiin ei myöskään ole kovin suuri eivätkä muutkaan sovelluskehitykset eroa toisistaan merkittävästi. Parhaimman sovelluskehityksen valinta muodostuu vielä monimutkaisemmaksi, mikäli sovelluksen on toimittava usealla eri alustalla. Valintaongelmaa onkin hyvä lähteä tarkastelemaan sekä yksittäisen sovelluksen tarpeiden että yrityksen sovellusympäristön kokonaistarpeiden näkökulmasta arvioiden esimerkiksi sovelluskehysvalinnan vaikutusta kehitettävien sovellusten elinkaarikustannuksiin.

9.2 Johtopäätökset

Tutkimus ei antanut selkeää vastausta mobiilin yritys ympäristön paikkatietosovellusten kehittämiseen käytettävän sovelluskehityksen valintaan. Tutkimus tarjosi kuitenkin joitain vastauksia. Esimerkiksi paikkatietosovellusten toiminnallisuustarpeiden näkökulmasta sovelluskehityksen valinnalla ei ole suurta merkitystä. Valinnassa onkin tärkeämpi keskittyä yrityksen tietojärjestelmäympäristön asettamiin tämän hetkisiin toiminnallisiin sekä pidemmän aikavälin suunniteltuihin tarpeisiin. Eräs olennainen valintaan vaikuttava tekijä on, millä mobiilialustoilla sovelluksia on tarkoitus käyttää.

Olennainen valintaa vaikeuttava tekijä on sovelluskehysten tulevaisuuden kehitysnäkymien ja kehityksen aikataulun epävarmuus, mistä esimerkkinä on tutkimuksen loppuvaiheilla SAP UI5:een tarjolle tullut alustava tuki mobiilikehitykselle, jonka tarkemmasta aikataulusta ei tutkimuksen alkuvaiheessa ollut vielä tietoa. Sovelluskehystä valitessa saatetaan joutua tekemään lukuisia oletuksia sovelluskehysten tulevaisuudesta. Tällöin etuna on, mikäli sovelluskehitykselle on luvattu sen kehittäjien puolesta tuki esimerkiksi seuraavaksi viideksi vuodeksi.

Sovelluskehysten tulevaisuuden näkymiä voidaan myös analysoida huomioiden, että sen tarjoamien toiminnallisuuksien määrä kasvanee jatkossa. Kehitysnäkymiin vaikuttaa

kuitenkin esimerkiksi sovelluskehityksen käyttäjäkunnan laajuus. Voidaan olettaa, että mitä laajempi käyttäjäkunta sovelluskehityksellä, sitä todennäköisemmin siihen kehitetään lisätoiminnallisuuksia, jotka saattavat olla myös yrityksen tarpeiden kannalta hyödyllisiä. (Robles ym. 2005) Sovelluskehysten kehittäjä- ja käyttäjämäärien arvioiminen voisikin tarjota lisäarvoa sovelluskehityksen valintaongelman tarpeiden näkökulmasta.

Tutkimuksen luotettavuuden kannalta lisäarvoa toisi myös, mikäli laadun arvioinnissa olisi käytetty useampaa henkilöä. Laadun arviointi on aina jollain tasolla subjektiivista, joten useamman henkilön toteuttama arviointi voisi tuoda esiin uusia näkökulmia ja myös mahdollisesti vähentää subjektiivisuuden vaikutusta arvioihin. Sovelluskehysten laatua voitaisiin arvioida myös eri näkökulmista tai eri kriteereillä, mikä voisi myös laajentaa analyysiä.

Laatuarviointeja voitaisiin myös toteuttaa eri menetelmillä, kuten sovelluskehysten käyttäjien haastatteluilta. Tällöin ongelmaksi saattaisi tosin muodostua käyttäjien vajavainen kokemus muista kuin eniten käyttämästään sovelluskehityksestä, mistä johtuen he saattaisivat puoltaa käyttämänsä sovelluskehitystä arvioinneissaan. Sovelluskehysten käyttäjien haastatteluihin verrattuna tämän tutkimuksen etuna onkin, että tekijällä ei ollut käytännössä ollenkaan aikaisempaa kokemusta mistään tutkimuksessa arvioidusta sovelluskehityksestä. Näin tutkimus voitiin toteuttaa ilman laajemmasta kokemuksesta mahdollisesti aiheutuvaa ennakko-oletusta tietyn sovelluskehityksen paremmuudesta.

Toisaalta tekijän vajavainen kokemus tutkimuksen sovelluskehityksistä yhdistettynä enimmäkseen pinnallisella tasolla toteutettuun arviointiin sovelluskehityksistä, ei anna kattavaa kuvaa niiden laadusta. Tutkimuksen tuloksia voidaan kuitenkin käyttää lähtökohtana mobiilin yritysympäristön paikkatietosovelluskehityksen valinnassa. Jatkona seuraavassa vaiheessa voitaisiin esimerkiksi toteuttaa sovelluskehityksillä prototyyppisovelluksia, jolloin sovelluskehityksiä ja niillä tehtävää kehitysprosessia voitaisiin analysoida syvällisemmin. Mahdollisesti tällöin voitaisiin hyödyntää myös määrällisiä mittareita laadun arvioinnissa, mikä voisi vähentää tutkimuksen subjektiivisuutta.

10 Yhteenveto

Työssä tutkittiin sovelluskehitystyökalun valintaa mobiilin paikkatietoa hyödyntävän yrityssovelluksen toteuttamiseksi SAP-ympäristössä. Kyseisen kontekstin sovellusten toteuttamiseen on tarjolla lukuisia vaihtoehtoisia sovelluskehityksiä ja tutkimuksen tärkeimpänä tavoitteena olikin mahdollistaa eri vaihtoehtojen vertailu. Vertailun tueksi luotiin ISO/IEC 9126 ohjelmistolaatumalliin pohjautuva kriteeristö, jonka perusteella sovelluskehitystyökalun laatua voidaan arvioida kyseisessä kontekstissa. Laatuarvion perusteella analysoitiin, mikä SAP:n tukemista mobiilisovelluskehityksistä on kokonaisuutena paras ja miten sovelluskehysten laatu ilmenee eri laatuksien tai ominaisuuksien osalta.

Yritysympäristön mobiileja paikkatietosovelluksia kehitettäessä täytyy usein huomioida tiettyjä sovellustarpeita ja -toiminnallisuuksia, kuten tietoturva, yhdistettävyyden ja paikannus. Onkin hyvä mikäli käytettävä sovelluskehitys tarjoaa ratkaisuja näiden huomioimiseksi. Yritysympäristön mobiilin tietotekniikan tarpeet myös muuttuvat ajan myötä, mistä syystä sovelluskehysten on hyvä mahdollistaa sovelluksen käyttäminen erilaisilla alustoilla. Monelle alustalle kehitettäessä tehdäänkin usein hybridi- tai web-sovelluskehitystä natiivikehityksen sijaan mahdollista näin saman sovelluskoodin käytön usealla alustalla. Haittapuolena on tällöin usein kuitenkin heikompi mobiililaitteen natiiviominaisuuksien hyödynnettävyys.

Tutkimuksen perusteella Sybase Mobile SDK tarjoaa parhaan ratkaisun SAP-ympäristön mobiilisovelluskehitykseen edellyttäen, että sovellus ajetaan vain yhdellä mobiilialustalla ja se voidaan asentaa mobiililaitteelle. Yrityksen muuttuvien mobiiliuden tarpeiden takia mobiilisovellusten on kuitenkin usein toimittava usealla alustalla. Tällöin Sybase Mobile SDK ei erotu enää edukseen, koska sillä natiivien sovellusten monialustakehitys on työläämpää kuin esimerkiksi Appcelerator:lla. Vaihtoehtoisesti usealle alustalle kehitettäessä voidaan kehittää hybridisovelluksia PhoneGap:llä, jos tarpeena on tuki mahdollisimman monelle alustalle, tai web-sovelluksia Sencha Touch:lla, jos sovellusta halutaan käytettävän ilman sen asentamista. Sybase Mobile SDK:n haittapuolena on myös, että sen lähdekoodi ei ole saatavilla.

PhoneGap, Appcelerator, Sencha Touch ja Sybase Mobile SDK soveltuvatkin kaikki SAP-ympäristön mobiilisovelluskehitykseen ja valinta niiden välillä riippuu sekä yksittäisten sovellusten tarpeista että yrityksen mobiilin tietotekniikka-arkkitehtuurin rakenteesta. Onkin ymmärrettävää, että SAP ei tue vain jotain tiettyä yksittäistä sovelluskehitystä, vaan se tukee useaa vaihtoehtoista sovelluskehitystä. Paikkatietoon liittyvien toiminnallisuuksien osalta sovelluskehitykset eivät eroa merkittävästi toisistaan. Sovelluskehityksiä voidaan käyttää myös yhdessä, millä voidaan poistaa niiden yksittäin käytettäessä ilmeneviä huonoja puolia. Tätä ei kuitenkaan tässä tutkimuksessa analysoitu tarkemmin johtuen erilaisten yhdistelmien suuresta määrästä.

Muista sovelluskehityksistä poiketen SAP UI5:ttä ei voida suositella yritysympäristön mobiilisovellustarpeiden täyttämiseen, koska sitä ei ole vielä tarkoitettu mobiilisovellusten toteuttamiseen. Se valittiin kuitenkin tutkimuksessa tarkasteltavaksi, koska sille on luvattu tuki mobiilisovelluskehitykselle. Tutkimuksen teon loppuvaiheilla siitä ilmestyi kehitysversio, joka sisältää erityisesti mobiilisovelluskehitykseen tarkoitettua toiminnallisuutta. SAP UI5:n hyödynnettävyys mobiilisovelluskehityksessä vaatii kuitenkin lisätutkimusta.

SAP UI5:lle tullut uusi versio, joka mahdollistaa mobiilisovelluskehityksen on hyvä esimerkki sovelluskehysten kehityksen nopeudesta. Kehityksen nopeus asettaakin haasteensa sovelluskehysten valinnalle, koska valintaa tehdessä on vertailtava sovelluskehysten nykyisiä ominaisuuksia huomioiden lisäksi mahdollisuuksien mukaan myös niiden tulevaisuuden kehityksen. Tutkimuksessa ei ole otettu kantaa sovelluskehysten tulevaisuuden kehitykseen, vaikka se onkin tärkeää sovelluskehysten valinnan kannalta.

Tutkimus ei anna yksiselitteistä vastausta, mikä sovelluskehys sopii parhaiten mobiilin yritys ympäristön paikkatietosovellustarpeiden täyttämiseen. Tarkempaa vertailua sovelluskehysten välillä voitaisiin tehdä esimerkiksi toteuttamalla sovelluskehyksillä pilottisovelluksia, joita voitaisiin vertailla hyödyntäen myös määrällisiä mittareita. Tutkimuksen merkittävin hyöty on kuitenkin, että se tarjoaa kontekstin ja kriteerit, joiden pohjalta sovelluskehysten vertailua voidaan lähestyä niin yksittäisen sovelluksen tarpeiden näkökulmasta kuin yrityksen mobiilisovellusarkkitehtuuria suunniteltaessa ja kehittäessä.

Lähdeluettelo

Aberdeen Group. (2000). The Benefits of Mobilizing Enterprise Applications with Handheld Devices.[Online]. [Viitattu 22.8.2012]. Saatavissa: http://www.paladin-ent.com/files/benefits_ent_apps.pdf.

Abowd, G. & Atkeson, C. & Hong, J. & Long, S. & Kooper, R. & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *Wireless Networks*. Vol 3:5. S. 421-433.

Adams, D.J. (2012). SAP UI5 - The Future direction of Sap UI Development? [Online]. [Viitattu 19.9.2012]. Saatavissa: <http://www.saportals.com/articles/share/36400/>.

Al-Mashari, M. & Al-Mudimigh, A. & Zairi, Mohamed. (2002). Enterprise resource planning: A taxonomy of critical factors. *European Journal of Operational Research*. Vol. 146:2. S. 352-364.

Al-Mashari, M. (2003). Enterprise resource planning (ERP) systems: a research agenda. *Industrial Management & Data Systems*. Vol 103:1. S. 22-27.

Appcelerator Inc. (2012). Appcelerator Titanium Mobile -dokumentaatio – versio 2.1. [Online]. [Viitattu 21.9.2012]. Saatavissa: <http://docs.appcelerator.com/titanium/2.1/index.html#>.

Apple. (2011). Introduction to Error Handling Programming Guide For COCOA. [Online]. iOS Developer Library. [Viitattu 6.12.2012]. Saatavissa: http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ErrorHandlingCocoa/ErrorHandling/ErrorHandling.html#//apple_ref/doc/uid/TP40001806-CH201-SW1.

Babar, M.A. & Zhu, L. & Jeffery, R. (2004). A framework for classifying and comparing software architecture evaluation methods, *Software Engineering Conference, 2004. Proceedings. 2004 Australian*. S. 309-318.

Ballagas, R. & Borchers, J. & Rohs, M. & Sheridan, J.G. (2006). The smart phone: a ubiquitous input device. *Pervasive Computing, IEEE*. Vol. 5:1. S. 70-77.

Barbará, D. (1999). Mobile Computing and Databases – A Survey. *IEEE Transactions on Knowledge and Data Engineering*. Vol. 11:1. S. 108-117.

Beck, Stefan. (2012). SAPUI5 version 1.8 is available. [Online]. SAP Community Network. [Viitattu 21.11.2012]. Saatavissa: <http://scn.sap.com/community/developer-center/front-end/blog/2012/11/14/sapui5-version-18-is-available>.

Bevis, D & Patterson, L. (2002). Extending Enterprise Applications to Mobile Users. *Websphere Everyplace Access*. IBM.

Botta-Genoulaz, V. & Millet, P. (2005). A classification for better use of ERP systems. *Computers in Industry*. Vol. 56:6. S. 573-587.

- Bradley, J. & Loucks, J. & Macaulay, J. & Medcalf, R. & Buckalew, L. (2012). BYOD: A Global Perspective. Harnessing Employee-Led Innovation. Kyselytutkimus. [Online] Cisco IBSG Horizons. [Viitattu 19.1.2013]. Saatavissa: http://resources.idgenterprise.com/original/AST-0074924_BYOD_Horizons-Global.pdf.
- Carrol, A & Heiser, G. (2010). An Analysis of Power Consumption in a Smartphone. Usenix 2010.
- Charland, A. & Leroux, B. (2011). Mobile application development: web vs. native. Commun.ACM, Vol. 54:5. S. 49-53.
- Chen, J.C.H. & Chiniwar, S. & Lin, B. & Chen, P. (2006). Security in e-business and beyond: a case study reflecting current situations and future trends. International Journal of Mobile Communications. Vol 4:1. S. 17-33.
- Davenport, T. (1998). Putting the enterprise into the enterprise system. Harvard Business Review. Vol. 76. S. 121-131.
- Densham, P. J. (1991). Spatial Decision Support Systems. Teoksessa: Maguire, D. J. & Goodchild M. S. & Rhind D. W. (eds) Geographical information systems: principles and applications, London: Longman. S. 403 - 412.
- Deveria, A. (2012). Can I use -sivusto. [Online]. [Viitattu 13.11.2012]. Saatavissa: <http://caniuse.com/>.
- ESRI. (2007). GIS Best Practices. Using GIS and SAP. [Online]. [Viitattu 9.11.2012]. Saatavissa: <http://www.esri.com/library/bestpractices/using-gis-and-sap.pdf>.
- Faulk, M. (2012). Mobile Frameworks Comparison Chart. [Online]. [Viitattu 19.9.2012] Saatavissa: <http://www.markus-falk.com/mobile-frameworks-comparison-chart/>.
- Fischer, A & Podgaetsky, G. (2012). Take Advantage of Cross-Platform, Cross-Device Access While Keeping Your Data Secure with SAP NetWeaver Gateway. [Online]. SAP insider. [Viitattu 13.11.2012]. Saatavissa: <http://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/c0d6853a-c7ac-2f10-f484-9a6ad818d511?QuickLink=index&overridelayout=true&55959128937473>.
- Forman, G.H. & Zahorjan, J. (1994). The challenges of mobile computing. IEEE Computer. Vol. 27:4. S. 38-47.
- Ganz, B. (2012). Get to Know the UI Development Toolkit for HTML5 (aka SAP UI5). [Online]. SAP Community Network. [Viitattu 4.12.2012]. Saatavissa: <http://scn.sap.com/docs/DOC-31625>.
- Google. (2011). Google APIs Terms of Service. [Online]. [Viitattu 12.1.2013]. Saatavissa: <https://developers.google.com/terms/>.
- Howe, J. (2006). The Rise of Crowdsourcing. [Online]. Wired. [Viitattu 22.8.2012]. Saatavissa: <http://www.wired.com/wired/archive/14.06/crowds.html>.

Hui, L. & Darabi, H. & Banerjee, P. & Jing, L. (2007). Survey of Wireless Indoor Positioning Techniques and Systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*. IEEE Transactions on. Vol. 37:6. S. 1067-1080.

Imielinski, T. & Badrinath, B.R. (1994). Mobile wireless computing: challenges in data management. *Commun.ACM*. Vol. 37:10. S. 18-28.

ISO 9241-11. (1998). Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability. Geneva, Sveitsi: International Organization for Standardization. 22s.

ISO 9241-210. (2010). Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems. Geneva, Sveitsi: International Organization for Standardization. 32s.

ISO/IEC 9126-1. (2001). Software engineering -- Product quality -- Part 1: Quality model. Geneva, Sveitsi: International Organization for Standardization. 25s.

ISO/IEC 14598-1. (1999). Information technology -- Software product evaluation -- Part 1: General overview. Geneva, Sveitsi: International Organization for Standardization. 19s.

ISO/IEC 14598-5. (1998). Information technology -- Software product evaluation -- Part 5: Process for evaluators. Geneva, Sveitsi: International Organization for Standardization. 35s.

ISO/IEC 25000. (2005). Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE. Geneva, Sveitsi: International Organization for Standardization. 41s.

Jacobs, R. & Weston, T. (2007). Enterprise resource planning (ERP)—A brief history. *Journal of Operations Management*. Vol 25:2. S. 357-363.

Kaluscha, O. (2011). SUP – why do we need Netweaver Mobile any longer?!? [Online]. Blogi. 5.10.2011. [Viitattu 19.9.2012]. Saatavissa: <http://www.placeworkers.com/sup-why-do-we-need-netweaver-mobile-any-longer/>.

Karakatsanis, J. (2010). Apple iPad. [Online]. [Viitattu 22.12.2012]. Saatavissa: <http://www.flickr.com/photos/johnkarakatsanis/4545524716/>.

Katz, E. (2012). How to Write Online Mobile Apps Consuming SAP Back End Systems. [Online]. Blogi. 12.8.2012. [Viitattu 25.9.2012] Saatavissa: <http://scn.sap.com/docs/DOC-30760?source=email-g-technology-news-newsletter-20120904>.

Korvenranta, H. (2005) Asiantuntija-arvioinnit. Ovaska, S., Aula, A. & Majaranta, P. (eds.) Käytettävyytutkimuksen menetelmät, 111-124. Tampereen yliopisto, Tietojenkäsittelytieteiden laitos B-2005-1.

Kraftig, D. & Banke, K. & Slama, D. (2004). Enterprise Soa: Service-Oriented Architecture Best Practices. Vol. 1. Hagerstown, MD, USA: Prentice Hall. 408s. ISBN: 0-13-146575-9

Kray, C. & Elting, C. & Laakso, K. & Coors, V. (2003). Presenting route instructions on mobile devices. Proceedings of the 8th international conference on Intelligent user interfaces 2003. ACM. S. 117-124.

Lehman, M.M. (1980). Programs, life cycles, and laws of software evolution. Proceedings of the IEEE. Vol. 68:9. S. 1060-1076.

Meng, L. & Zipf, A. & Reichenbacher, T. (eds). (2004). Map-based mobile services: theories, methods, and implementations. Springer Berlin.

Moy, J. (2011). Can we embrace the Long Haired developer.[Online]. SAP for Mobile, Blogi. 3.11.2011. [Viitattu 7.11.2012]. Saatavissa: <http://scn.sap.com/community/mobile/blog/2011/11/03/can-we-embrace-the-long-haired-developer>.

Moy, J. (2012). Plumbing SAP's mobile architecture. [Online]. SAP for Mobile, Blogi. 20.4.2012. [Viitattu 17.8.2012]. Saatavissa: <http://scn.sap.com/community/mobile/blog/2012/04/20/plumbing-saps-mobile-architecture>.

Natchetoi, Y. & Kaufman, V. & Shapiro, A. (2008). Service-oriented architecture for mobile applications. Proceedings of the 1st international workshop on Software architectures and mobility 2008. ACM. S. 27-32.

Nielsen, J. (2006). UsabilityNet: Heuristic Evaluation. [Online]. [Viitattu 28.9.2012]. Saatavissa: <http://www.usabilitynet.org/tools/expertheuristic.htm>.

Oinas-Kukkonen, H. & Kurkela, V. (2003). Developing Successful Mobile Applications, Computer Science and Technology. *CST 2003*.

Panorama Consulting Group. (2011). 2011 Guide to ERP Systems and Vendors. [Viitattu 7.11.2012]. Saatavissa: <http://panorama-consulting.com/Documents/2011-Guide-to-ERP-Systems-and-Vendors.pdf>.

PhoneGap. (2012). PhoneGap | Supported Features. [Online]. [Viitattu 23.10.2012]. Saatavissa: <http://www.phonegap.com/about/feature>.

Reichenbacher, T. (2001). Adaptive concepts for a mobile cartography. Journal of Geographical Sciences. Vol. 11:0. S. 43-53.

Reichenbacher, T. (2004). Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices. Väitöskirja. Institute of Photogrammetry und Cartography, Technical University, Munich.

Robles, G. & Amor, J.J. & Gonzalez-Barahona, J.M. & Herraiz, I. (2005). Evolution and growth in large libre software projects, Principles of Software Evolution, Eighth International Workshop on 2005. S. 165-174.

Sancho85. (2012). Galaxy Nexus, Wikipedia. [Online]. [Viitattu 22.12.2012]. Saatavissa: http://en.wikipedia.org/wiki/File:Galaxy_nexus.jpg#filelinks.

Sanjeet, M. & Tzanko, S. & Stanley, S. (2012). Mobilizing Your Enterprise with SAP. Boston, USA: SAP Press. S. 409. ISBN: 978-1-59229-419-0.

SAP. (2012a). SAP Newsroom. [Online]. [Viitattu 17.7.2012]. Saatavissa: <http://www.news-sap.com/>.

SAP. (2012b). SAP Fact Sheet October 2012. [Online]. [Viitattu 7.11.2012]. Saatavissa: <http://www.sap.com/corporate-en/investors/pdf/SAP-Fact-Sheet-EN.pdf>.

SAP. (2012c). SAP Newsroom. SAP Drives Openness and Choice for Millions of Mobile App Developers. [Online]. [Viitattu 19.9.2012]. Saatavissa: <http://www.news-sap.com/sap-drives-openness-and-choice-for-millions-of-mobile-app-developers/>

Satyanarayanan, M. (1996). Fundamental challenges in mobile computing. Teoksessa: Burns, E & Moses, Y. Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing 1996. Philadelphia, PA, USA: ACM. S. 1-7. ISBN: 0-89791-800-2.

Schilit, B. & Adams, N. & Want, R. (1994). Context-Aware Computing Applications, Mobile Computing Systems and Applications. WMCSA 1994. First Workshop on 1994. S. 85-90.

Senna, R. (2005). Célulár. [Online]. [Viitattu 22.12.2012]. Saatavissa: <http://www.flickr.com/photos/negativz/38422354/>.

Soh C. & Kien, S.S. & Tay-Yap, J. (2000). Enterprise resource planning: cultural fits and misfits: is ERP a universal solution? Commun.ACM. Vol. 43:4. S. 47-51.

Statista. (2012). Global mobile applications user base forecast 2010-2015. [Online]. [Viitattu 6.11.2012]. Saatavissa: <http://www.statista.com/statistics/219959/global-mobile-applications-user-base-forecast/>.

Strategy Analytics. (2012). Worldwide Smartphone Population Tops 1 Billion in Q3 2012. [Online]. [Viitattu 6.11.2012]. Saatavissa: <http://blogs.strategyanalytics.com/WDS/post/2012/10/17/Worldwide-Smartphone-Population-Tops-1-Billion-in-Q3-2012.aspx>.

Sybase. (2011). What's the Point? A Comparison of Middleware versus Point Solutions. [Online]. [Viitattu 18.9.2012]. Saatavissa: http://www.sybase.com/files/White_Papers/Sybase_MobileMiddleware_vs_PointSolutions_wp.pdf [9/18, 2012].

Sybase. (2012). Sybase Unwired Platform 2.1 - Sybase Mobile SDK. [Online]. [Viitattu 25.9.2012] Saatavissa: <http://infocenter.sybase.com/help/index.jsp?topic=/com.sybase.infocenter.dc01204.0210/doc/html/koh1314329652870.html>.

Taniar, D. & Rahayu, J. (2002). Parallel sort-merge object-oriented collection join algorithms. Comput Syst Sci Eng. Vol. 17:3. S. 145-158.

Thomson, G. (2012). BYOD: enabling the chaos. Network Security. Vol 2. S. 5-8.

- van Berkel, C.H. (2009). Multi-core for mobile phones, Proceedings of the Conference on Design, Automation and Test in Europe 2009, European Design and Automation Association. S. 1260-1265.
- Virrantaus, K. & Markkula, J. & Garmash, A. & Terziyan, V. & Veijalainen, J. & Katanosov, A. & Tirri, H. (2001). Developing GIS-supported location-based services. Web Information Systems Engineering, 2001. Proceedings of the Second International Conference on 2001. Vol 2. S. 66-75.
- W3C. (2010). Web SQL Database. W3C Working Group Note 18.10.2010. [Online]. [Viitattu 13.11.2012]. Saatavissa: <http://www.w3.org/TR/webdatabase/>.
- W3C. (2011). Web Storage. W3C Candidate Recommendation 8.12.2011. [Online]. [Viitattu 13.11.2012]. Saatavissa: <http://www.w3.org/TR/webstorage/>.
- W3C. (2012a). HTML5. W3C Editor's Draft 21.11.2012. [Online]. [Viitattu 27.11.2012]. Saatavissa: <http://dev.w3.org/html5/spec/single-page.html>.
- W3C. (2012b). DeviceOrientation Event Specification. W3C Editor's Draft 13.6.2012. [Online]. [Viitattu 6.11.2012]. Saatavissa: <http://dev.w3.org/geo/api/spec-source-orientation.html#deviceorientation>.
- W3C. (2012c). Geolocation API Specification. W3C Proposed Recommendation 10.5.2012. [Online]. [Viitattu 31.10.2012]. Saatavissa: <http://www.w3.org/TR/geolocation-API/>.
- Walter, T. & Bussard, L. & Roudier, Y. & Haller, J. & Kilian-Kehr, R. & Posegga, J. & Robinson, P. (2004). Secure mobile business applications – framework, architecture and implementation. Information Security Technical Report. Vol. 9:4. S. 6-21.
- Waluyo, A. & Srinivasan, B. & Taniar, D. (2005). Research on location-dependent queries in mobile databases. Int J Comput Syst: Sci Eng. Vol. 20:3. S. 77–93.
- Waluyo, A. & Taniar, D. & Rahayu, W. & Srinivasan, B. (2009). Mobile service oriented architectures for NN-queries. Journal of Network and Computer Applications. Vol. 32:2. S. 434-447.
- Watson, T. (1996). Application Design for Wireless Computing. Teoksessa: Imielinski, T. & Korth, H.F. (eds.) Mobile Computing. Springer US. S. 363-373.
- Wong, B. & Jeffery, R. (2001). Cognitive Structures of Software Evaluation: A Means-End Chain Analysis of Quality. Teoksessa: F. Bomarius & S. Komi-Sirviö, (eds), Product Focused Software Process Improvement. Springer Berlin Heidelberg. S. 6-26.